Spring Cloud Eureka:

**🔍 ¿Qué hace?**

* Lleva un **registro dinámico** de qué microservicios están activos, dónde están y en qué puerto/IP.
* Permite que otros servicios **descubran** a estos microservicios **por nombre**, sin preocuparse por direcciones físicas.

Es decir, En una arquitectura de **microservicios**, tienes muchas aplicaciones que se ejecutan en diferentes lugares (puertos, servidores, contenedores, etc.).

**Eureka Server** funciona como una “guía telefónica” que lleva el registro de todos los microservicios disponibles.

**💡 ¿Y cómo lo consumen otros servicios?**

* Directamente usando librerías como:
  + RestTemplate con @LoadBalanced
  + ✅ OpenFeign con @FeignClient(name = "nombre-del-servicio")
  + Spring Cloud Gateway (para ruteo)

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Pasos:

Primero creas un microservicio llamado **eureka-server**

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.  
Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Y en el appliction.yml pon esto:  
Texto

El contenido generado por IA puede ser incorrecto.  
**Listo ya tienes eureka server, ahora solo falta correrlo:**

Debes correrlo en el puerto que definiste en este caso **el localhost:8761**  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

**Aun no hay microservicios registrados, para eso, primero debes crear microservicios rápidos:**

Creas un microservicio llamado **eureka-client**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Su application.yml

Texto

El contenido generado por IA puede ser incorrecto.

Luego lo corres con mvn spring-boot:run

Le das primero al puerto 8081 si el rest se cargo bien:  
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Luego vas al eureka server para verificar si te carga el puerto. Si te carga, te saldrá esto:  
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Ya aparece en el puerto, ahora si le das click al link te saldrá esto

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Te saldrá esto porque es el info que tu definiste en tu app.yml. Ojo no es igual al rest, es diferente. Una cosa es el rest donde puedes ver la consulta, y otro es ver el estado de tu aplicación.

Esto funciona con el **Spring boot actuator,** es opcional, el actuator lo que hace es mostrar. Ahí va una explicación:  
**⚙️ ¿Qué es Spring Boot Actuator?**

Actuator es una **herramienta de monitoreo y administración** de aplicaciones Spring Boot.

Te da acceso a endpoints especiales como:

|  |  |
| --- | --- |
| **Endpoint** | **¿Qué hace?** |
| /actuator/health | Muestra si la app está "UP" (funcionando) |
| /actuator/info | Muestra información del app (nombre, versión, etc.) |
| /actuator/metrics | Estadísticas de uso (memoria, CPU, etc.) |
| /actuator/env | Muestra las variables de configuración |
| /actuator/beans | Lista todos los beans de Spring cargados |

**🧠 ¿Para qué sirve en la práctica?**

* 📊 Ver si un microservicio está funcionando (/health)
* ℹ️ Mostrar información útil en Eureka (/info)
* 🔍 Detectar errores más fácilmente (ver configs, métricas)
* 🔒 Proteger endpoints sensibles (se pueden asegurar)

**🧪 ¿Cómo trabajan juntos Eureka y Actuator?**

Spring Boot Actuator **complementa** a Eureka al permitir que:

* Eureka revise /actuator/health para saber si un servicio sigue vivo
* La consola de Eureka muestre datos de /actuator/info al hacer clic en una instancia

👉 Si el servicio no responde a /actuator/health, Eureka lo marca como DOWN

**🎯 Resumen Final**

|  |  |
| --- | --- |
| **Herramienta** | **¿Para qué sirve?** |
| **Eureka** | Registro y descubrimiento de microservicios |
| **Actuator** | Monitoreo, métricas y endpoints especiales del sistema |