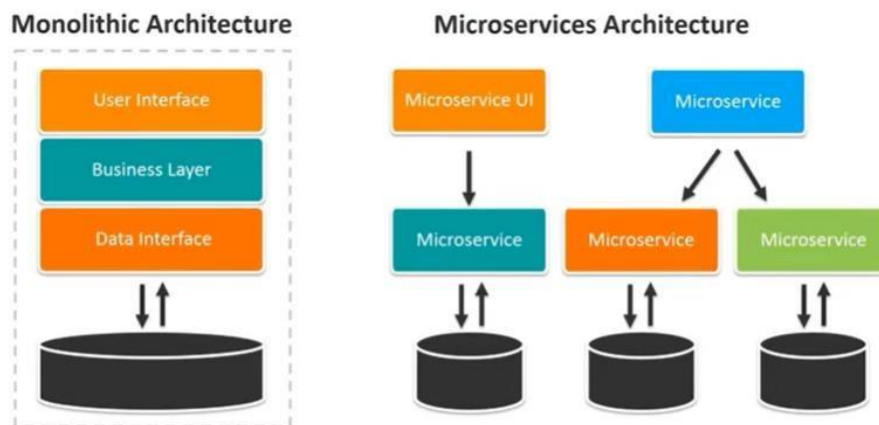


Microservicios con Spring Boot y Spring Cloud

¿Qué es una aplicación monolítica?

El enfoque tradicional para el diseño de aplicaciones se centraba en la arquitectura monolítica, en que todos los elementos que pueden implementarse estaban contenidos en una sola aplicación. Este enfoque tiene sus desventajas: cuanto más grande es la aplicación, más difícil es solucionar los problemas que se presentan y agregar funciones nuevas rápidamente. En cambio, si las aplicaciones se diseñan con microservicios, se resuelven dichos problemas y se impulsa el desarrollo y la capacidad de respuesta.



¿Para qué sirven los microservicios?

Los microservicios son tanto un estilo de arquitectura como un modo de programar software. Con los microservicios, las aplicaciones se dividen en sus elementos más pequeños e independientes entre sí. A diferencia del enfoque tradicional y monolítico de las aplicaciones, en el que todo se compila en una sola pieza, los microservicios son elementos independientes que funcionan en conjunto para llevar a cabo las mismas tareas. Cada uno de esos elementos o procesos es un microservicio.



¿Qué es Spring Boot?

Spring Boot es un módulo del proyecto de Spring que fue creado para simplificar el desarrollo de aplicaciones con Spring Framework bajo licencia Apache 2.0.

¿Que es Spring Cloud?

El framework está construido sobre Spring Boot, el cual proporciona auto-configuraciones básicas para facilitar el desarrollo de los microservicios, además de proporcionar ya un contenedor de aplicaciones (Tomcat, Jetty, Undertow) embebido.

¿Qué es Feign Client?

Feign es una librería que forma parte del stack de Spring Cloud, desarrollada por Netflix, para generar clientes de servicios REST de forma declarativa.

Feign ha sido creado para facilitar la integración entre microservicios mediante la creación de clientes HTTP de forma declarativa. Esto significa que, se simplifica tanto la creación de clientes, que el desarrollador simplemente necesita anotar una interfaz para tenerlo.

¿Qué es Rest Template?

REST Template es la clase que ofrece Spring para el acceso desde la parte cliente a Servicios REST. Conceptualmente sería el equivalente al JdbcTemplate o al JmsTemplate.

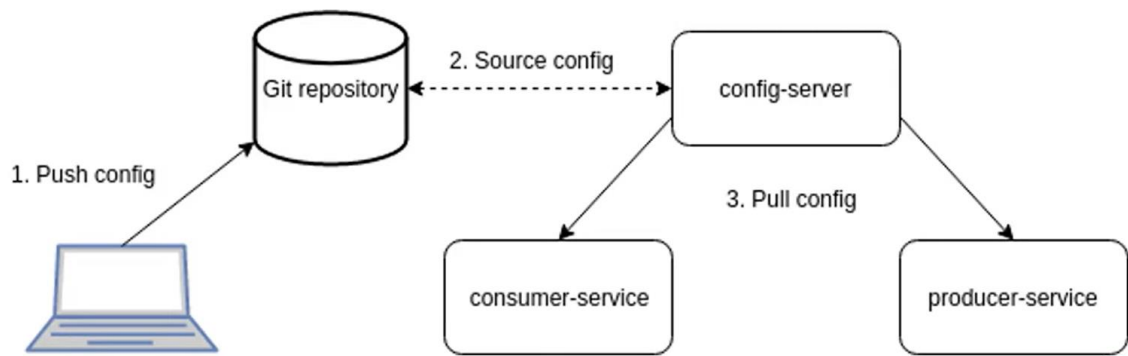
¿Qué es el servidor de configuración?

En arquitecturas distribuidas, con múltiples microservicios desplegados en distintos servidores, realizar cambios de configuración en cada una de ellos puede tornarse una tarea compleja y demandante.

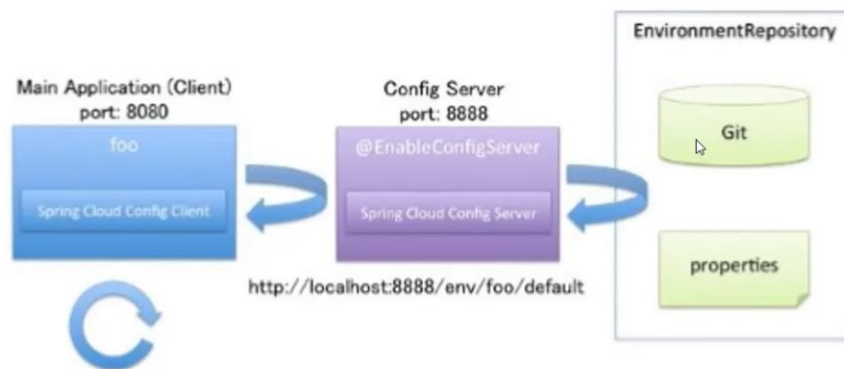
Spring Cloud Config es una herramienta que permite exteriorizar y centralizar la configuración de los microservicios en un solo lugar, lo que facilita enormemente la administración de la configuración del sistema.

Por default, Spring Cloud Config utiliza Git como repositorio para centralizar la configuración de cada uno de los servicios. Esto nos permite aprovechar algunas de sus ventajas:

- Control de cambios
- Utilizar labels para hacer referencia a versiones específicas



Su función es almacenar las propiedades de configuración de los microservicios del ecosistema

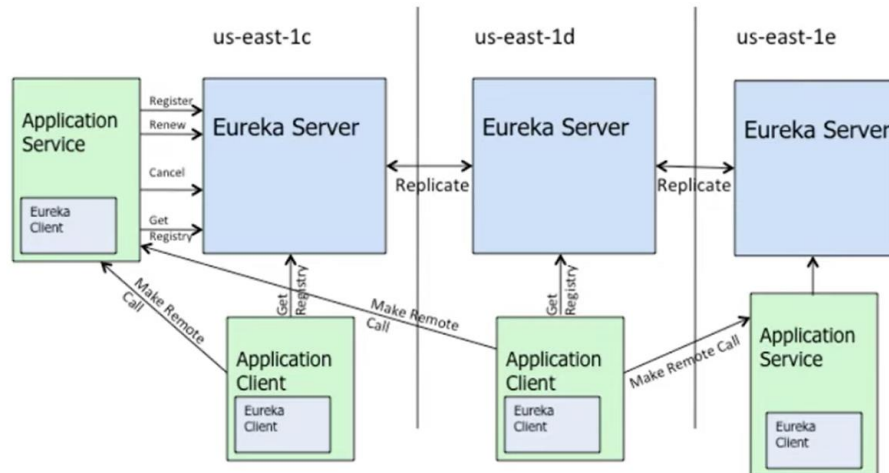


Cloudconfig se integra con Eureka de forma que, durante el arranque de un microservicio, si este es un Eureka client consultará al servidor Eureka donde se encuentra el servidor de configuración y le solicitará su configuración.

Al almacenar la configuración en un repositorio git esto proporciona además de un repositorio centralizado de configuración, un histórico del mismo. Además, cloudconfig añade funcionalidades a esta integración con git permitiendo consultar configuración de diferentes ramas.

¿Qué es Eureka?

Eureka es un servidor para el registro y localización de microservicios, balanceo de carga y tolerancia a fallos. La función de Eureka es registrar las diferentes instancias de microservicios existentes, su localización, estado, etc.



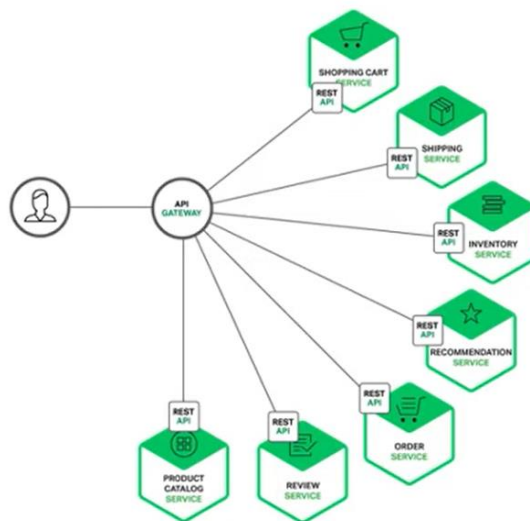
Abstracción de la localización física de los microservicios: cualquier microservicio que sea un cliente Eureka solo necesita conocer el identificador del microservicio al que desea invocar y Eureka resolverá su localización, puerto, etc.

Conocer el estado de nuestro ecosistema de microservicios. Eureka proporciona un dashboard que permite ver los microservicios existentes actualmente en el registro.

¿Qué es Gateway?

Es una puerta de enlace que nos da acceso a cada microservicio.

API GateWay Pattern o patrón de pasarela de API. Este patrón se encarga de definir un nuevo API a partir de las APIs existentes que enfoque de una manera más directa una casuística determinada.



¿Qué es circuit breaker?

El patrón Circuit Breaker evita que una aplicación intente de manera reiterada una operación que con probabilidad vaya a fallar, permitiendo que esta continúe con su ejecución sin malgastar recursos mientras el problema no se resuelva.

¿Qué es Sleuth?

Spring Cloud Sleuth proporciona la infraestructura para que las peticiones salientes envíen un identificador de correlación de la petición global y para las peticiones entrantes relacionarlo con la petición global.

¿Qué es Zipkin?

Es una herramienta de rastreo de microservicios. Zipkin es una herramienta que recolecta las transacciones creadas por Sleuth en la ejecución de los microservicios e información de los tiempos de respuesta de las invocaciones que han intervenido en una transacción. Ofrece las dos funcionalidades la recolección de datos y la obtención de los mismos.

<https://zipkin.io/pages/quickstart.html>