04

La configuration centralisée

Centralisez et dynamisez la configuration de vos microservices

La Prolifération des fichiers de configuration

Dans une architecture microservices, chaque service a sa propre configuration (application.yml, application.properties).

Scénario:

- **Redondance :** Les mêmes propriétés sont dupliquées dans des dizaines de fichiers.
- **Incohérence :** Une modification nécessite de modifier, de re-compiler et de re-déployer *tous* les services concernés. Le risque d'erreur est énorme.
- **Gestion des environnements :** Gérer les différences entre les environnements dev, qa, et prod devient un casse-tête de profils Spring et de fichiers multiples.
- **Sécurité :** Des informations sensibles (mots de passe, clés d'API) sont disséminées dans de nombreux dépôts de code.

— Question:

Comment gérer la configuration de manière centralisée, versionnée et sécurisée, sans avoir à redémarrer les services pour chaque changement ?

Spring Cloud Config Server

Spring Cloud Config propose une approche client-serveur pour externaliser la configuration.

Le config server

Un microservice Spring Boot dédié dont le seul rôle est de fournir des propriétés de configuration à d'autres applications.

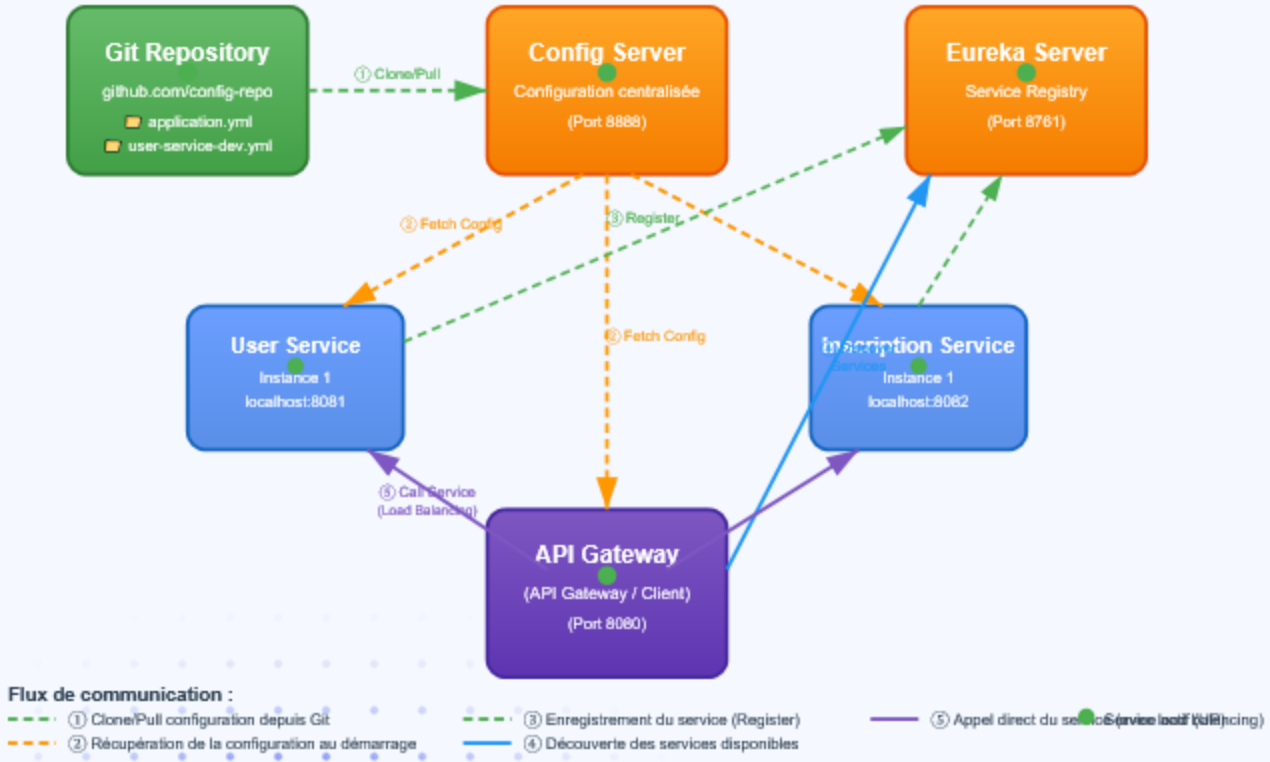
Les clients config

Les microservices sont configurés pour contacter le Config Server au démarrage afin de récupérer leur configuration.

Le référentiel

Le Config Server ne stocke pas la configuration lui-même.
Il la lit depuis un référentiel externe.

Spring Cloud Config Server



Spring Cloud Config Server: Architecture

1. Au démarrage, le microservice client contacte le Config Server en s'identifiant(avec son nom d'application (`spring.application.name`).
2. Le Config Server reçoit la demande, clone le dépôt Git (s'il ne l'a pas déjà fait) ou le met à jour.
3. Il assemble les fichiers de configuration pertinents (voir diapositive suivante) et renvoie un ensemble de propriétés au client.
4. Le client reçoit ces propriétés et les injecte dans son contexte Spring, avec une priorité plus élevée que ses propres fichiers application.yml locaux.

Stockage de fichiers de configurations

Spring Cloud Config prend en charge le stockage des fichiers de configuration dans un certain nombre de backends différents:

- Répertoire Git, par exemple, sur Github ou Bitbucket
- Système de fichier local
- Base de données relationnelle
- HashiCorp Vault