Video 19

**CENTRALIZED CONFIGURATION**

Centralized configuration dalam konteks arsitektur microservice adalah pendekatan di mana semua konfigurasi aplikasi disimpan dan dikelola di satu lokasi pusat. Ini memungkinkan semua layanan mikro (microservices) untuk mengakses dan mengelola konfigurasi mereka dari satu sumber, sehingga memudahkan pengelolaan, konsistensi, dan perubahan konfigurasi secara real-time.

**Keuntungan Centralized Configuration**

1. **Konsistensi**:
   * Semua layanan menggunakan satu sumber konfigurasi yang sama, memastikan bahwa konfigurasi tetap konsisten di seluruh layanan.
2. **Pengelolaan Mudah**:
   * Memungkinkan administrator untuk mengelola konfigurasi secara terpusat, mengurangi kerumitan pengelolaan konfigurasi di setiap layanan secara individu.
3. **Kemampuan Adaptasi**:
   * Memungkinkan perubahan konfigurasi diterapkan secara real-time tanpa perlu me-restart layanan, sehingga meningkatkan fleksibilitas dan kecepatan respons terhadap perubahan.
4. **Keamanan**:
   * Konfigurasi sensitif seperti kredensial dapat dienkripsi dan disimpan dengan aman di satu lokasi pusat.
5. **Versi dan Riwayat**:
   * Dapat menyimpan versi sebelumnya dari konfigurasi, memudahkan rollback jika ada masalah dengan konfigurasi terbaru.

**Contoh Implementasi Centralized Configuration**

1. **Spring Cloud Config**:
   * Framework yang menyediakan solusi centralized configuration untuk aplikasi Spring.
   * Mendukung konfigurasi yang diambil dari berbagai sumber seperti Git, SVN, dan file sistem lokal.
   * Konfigurasi dapat diakses melalui REST API yang disediakan oleh Spring Cloud Config Server.
2. **Consul**:
   * Selain sebagai service registry, Consul juga menyediakan fitur untuk centralized configuration.
   * Menggunakan key-value store untuk menyimpan konfigurasi yang dapat diakses oleh semua layanan.
3. **Etcd**:
   * Store key-value distribusi yang banyak digunakan dalam ekosistem Kubernetes untuk menyimpan konfigurasi dan metadata aplikasi.
   * Mendukung konfigurasi dinamis dan berbasis event.
4. **Zookeeper**:
   * Digunakan untuk penyimpanan konfigurasi dalam beberapa sistem besar yang memerlukan konsistensi data tinggi.
   * Mendukung watch pada perubahan data sehingga layanan dapat bereaksi terhadap perubahan konfigurasi.