Tugas Microservices

Nama : Muhammad Raihan
NIM : 2301083018
Kelas : TK2 A

1. Jelaskan Tentang Server side Discovery
2. Jelaskan Tentang Cleint side Discovery
3. Jelaskan kekurangan dan kelebihan dari no 1 \dan 2 diatas
4. Jelaskan tentang Service Registry

1. Server-side Discovery

Server-side Discovery adalah metode pencarian layanan dalam sistem terdistribusi di mana klien mengirim permintaan ke load balancer atau service discovery server, yang kemudian menentukan instance layanan yang tersedia dan meneruskan permintaan ke layanan yang sesuai.

Cara Kerja Server-side Discovery:

1. Klien mengirimkan permintaan ke load balancer atau service discovery server.
2. Load balancer/service discovery akan mengecek daftar layanan yang terdaftar dalam **Service Registry**.
3. Load balancer/service discovery memilih instance layanan yang sesuai berdasarkan kebijakan tertentu (misalnya round-robin, least connections, atau weighted routing).
4. Load balancer meneruskan permintaan klien ke instance layanan yang dipilih.

Contoh Implementasi:

1. AWS Elastic Load Balancer (ELB)
2. Kubernetes Service (kube-proxy)
3. Nginx atau HAProxy sebagai load balancer

2. Client-side Discovery

Client-side Discovery adalah metode di mana klien sendiri yang bertanggung jawab untuk menemukan dan memilih instance layanan berdasarkan daftar layanan yang tersedia dari **Service Registry**.

Cara Kerja Client-side Discovery:

1. Klien menghubungi **Service Registry** untuk mendapatkan daftar instance layanan yang tersedia.
2. Klien menerapkan algoritma load balancing untuk memilih instance layanan yang paling sesuai.
3. Klien langsung mengirimkan permintaan ke instance layanan yang dipilih.

Contoh Implementasi:

1. Netflix Eureka (Spring Cloud Netflix)
2. Consul
3. Zookeeper

3. Kelebihan dan Kekurangan Server-side Discovery vs Client-side Discovery

Faktor	Server-side Discovery	Client-side Discovery
Kelebihan	- Tidak membebani klien dengan tugas discovery. - Skalabilitas lebih baik karena menggunakan load balancer. - Cocok untuk sistem berbasis microservices di cloud (AWS, Kubernetes).	- Mengurangi dependensi pada load balancer. - Latensi lebih rendah karena klien berkomunikasi langsung dengan layanan tanpa perantara. - Memberikan fleksibilitas bagi klien dalam memilih strategi load balancing.
Kekurangan	- Memerlukan infrastruktur tambahan seperti load balancer. - Load balancer bisa menjadi single point of failure.	- Klien harus memiliki logika tambahan untuk discovery dan load balancing. - Setiap klien harus memiliki informasi yang up-to-date tentang layanan yang tersedia.

4. Service Registry

Service Registry adalah komponen dalam arsitektur microservices yang bertindak sebagai database pusat untuk menyimpan informasi tentang instance layanan yang tersedia. Service Registry memungkinkan layanan untuk **mendaftarkan diri**, **diperbarui**, dan **diambil informasinya** oleh klien atau load balancer.

Fungsi Service Registry:

1. **Pendaftaran Layanan**: Saat layanan baru di-deploy, ia mendaftarkan dirinya ke registry.
2. **Pembaruan Layanan**: Layanan yang berjalan dapat memperbarui statusnya (misalnya menambahkan atau menghapus instance).
3. **Penghapusan Layanan**: Jika layanan mati atau tidak tersedia, instance akan dihapus dari registry.
4. **Query untuk Discovery**: Klien atau load balancer dapat melakukan query ke registry untuk menemukan layanan yang tersedia.

Contoh Service Registry:

1. **Netflix Eureka**: Digunakan dalam ekosistem Spring Cloud.
2. **Consul**: Mendukung health checking dan konfigurasi service discovery.
3. **Zookeeper**: Digunakan oleh Apache Kafka dan Hadoop.

Kesimpulan

1. **Server-side Discovery** lebih cocok untuk arsitektur berbasis cloud dan skala besar karena menggunakan load balancer sebagai perantara.
2. **Client-side Discovery** lebih efisien dalam hal latensi tetapi membutuhkan klien yang lebih kompleks.
3. **Service Registry** adalah komponen penting dalam service discovery yang memastikan layanan dapat ditemukan dengan mudah dalam ekosistem microservices.