**Nama : Deya Mirza**

**NIM : 2301083009**

**Matkul : Microservice**

**Tugas**

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

**Jawab**

1. **Server-Side Discovery** adalah metode di mana **load balancer atau service registry** bertanggung jawab untuk menentukan layanan mana yang akan menangani permintaan dari klien. Dalam model ini, klien hanya mengirimkan permintaan ke satu endpoint tetap, dan server akan meneruskannya ke layanan yang sesuai berdasarkan informasi yang tersedia.

**Cara kerja**

* Klien mengirimkan permintaan ke load balancer atau service registry.
* Load balancer/service registry memeriksa layanan yang tersedia.
* Permintaan diteruskan ke layanan yang sesuai berdasarkan algoritma tertentu (round-robin, least connections, dll.).
* Layanan yang menerima permintaan mengolahnya dan mengembalikan respons ke klien.

**Contoh Implementasi**

* AWS Elastic Load Balancing (ELB)
* Kubernetes Service (ClusterIP, NodePort, LoadBalancer)
* NGINX atau HAProxy sebagai reverse proxy

1. **Client-Side Discovery** adalah metode di mana klien sendiri bertanggung jawab untuk menemukan dan memilih layanan yang akan digunakan. Dalam model ini, klien menghubungi service registry untuk mendapatkan daftar layanan yang tersedia, lalu langsung mengirim permintaan ke salah satu layanan tersebut.

**Cara kerja**

* Klien meminta daftar layanan yang tersedia dari service registry.
* Klien memilih satu layanan berdasarkan strategi tertentu (misalnya round-robin atau weighted load balancing).
* Klien mengirimkan permintaan langsung ke layanan yang dipilih.
* Layanan mengolah permintaan dan mengirimkan respons kembali ke klien.

**Contoh Implementasi**

* Netflix Eureka
* Consul
* Zookeeper
* Spring Cloud LoadBalancer

1. Kelebihan & Kekurangan Server-Side vs. Client-Side Discovery

|  |  |  |
| --- | --- | --- |
| Aspek | Server-Side Discovery | Client-Side Discovery |
| Kelebihan | * Klien tetap sederhana, tidak perlu logic pemilihan layanan. * Load balancer dapat melakukan pengelolaan trafik dengan lebih baik. * Cocok untuk lingkungan dengan skala besar. | * Tidak memerlukan load balancer eksternal, mengurangi bottleneck. * Klien memiliki kendali penuh atas pemilihan layanan, memungkinkan optimasi latensi. |
| Kekurangan | * Membutuhkan load balancer/service proxy tambahan yang bisa menjadi single point of failure (SPOF). * Meningkatkan latensi karena permintaan harus melewati load balancer. | * Klien harus mengelola discovery logic sendiri. * Kompleksitas bertambah karena setiap klien harus bisa menangani perubahan daftar layanan secara dinamis. |

1. **Service Registry** adalah komponen pusat yang menyimpan daftar layanan yang tersedia dalam suatu sistem terdistribusi. Service registry membantu klien dan layanan lain untuk menemukan layanan yang aktif dan dapat digunakan.

**Fungsi**

* Menyimpan dan Memperbarui Daftar Layanan → Layanan yang aktif mendaftarkan diri ke registry.
* Menyediakan Informasi Layanan → Klien atau load balancer dapat mencari layanan yang tersedia.
* Menyediakan Health Check → Registry bisa mengecek apakah layanan masih hidup atau tidak.

**Contoh**

* Netflix Eureka → Digunakan dalam ekosistem Spring Cloud.
* Consul → Mendukung health checks bawaan dan DNS-based discovery.
* Apache Zookeeper → Banyak digunakan dalam sistem berbasis Hadoop dan Kafka.
* Etcd → Digunakan dalam Kubernetes untuk menyimpan data cluster.