Nama: SYFA AULIA

Kelas: TK 2A

Matkul: Microservice

1. Jelaskan Tentang Server-side Discovery

Server-side discovery adalah metode dalam service discovery di mana client tidak langsung mencari lokasi layanan yang dibutuhkan, tetapi mengandalkan Load Balancer atau Service Registry yang berjalan di sisi server untuk menangani permintaan dan meneruskannya ke instance layanan yang sesuai.

Cara Kerja:

* Service instances (misalnya, layanan mikro dalam arsitektur microservices) mendaftarkan diri ke Service Registry.
* Saat client ingin mengakses layanan, permintaan dikirim ke Load Balancer atau Service Discovery Server.
* Server akan mencari layanan yang tersedia di Service Registry dan meneruskan permintaan ke instance yang sesuai.

Contoh Teknologi:

* Netflix Eureka (dengan Ribbon)
* Consul
* AWS Elastic Load Balancing (ELB)
* Kubernetes Service (ClusterIP, LoadBalancer)

1. Jelaskan Tentang Cleint side Discovery

Client-side discovery adalah metode di mana client sendiri yang bertanggung jawab untuk mencari dan menentukan ke mana permintaan harus dikirim. Client melakukan query ke Service Registry untuk mendapatkan daftar instance layanan yang tersedia dan memilih salah satunya.

Cara Kerja:

* Service instances mendaftarkan diri ke Service Registry.
* Client mengakses Service Registry untuk mendapatkan daftar layanan yang aktif.
* Client memilih instance layanan yang sesuai (misalnya dengan round-robin atau load balancing sendiri) dan langsung mengirimkan permintaan ke instance tersebut.

Contoh Teknologi:

* Netflix Eureka
* Consul
* ZooKeeper
* Kubernetes DNS-based Service Discovery

1. Jelaskan kekurangan dan kelebihan dari no 1 \dan 2 diatas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Metode** | | |  | | --- | | **Kelebihan** | | |  | | --- | | **Kekeurangan** | |
| Server-Side Discovery | * Beban load balancing ditangani oleh server/service registry, sehingga client lebih sederhana. * Cocok untuk arsitektur yang menggunakan API Gateway atau load balancer. * Client tidak perlu mengetahui lokasi langsung dari layanan. | * Membutuhkan infrastruktur tambahan (load balancer/service discovery server). * Bisa menjadi single point of failure jika tidak ada replikasi. |
| Client-Side Discovery | * Tidak memerlukan load balancer tambahan, karena client langsung berkomunikasi dengan service instance. * Mengurangi latensi, karena tidak ada perantara tambahan. * Skalabilitas lebih tinggi karena client menangani distribusi trafik. | * Client harus lebih kompleks karena perlu mengatur mekanisme load balancing dan service discovery. * Bisa terjadi thundering herd problem jika banyak client meminta service registry dalam waktu bersamaan. |

1. Jelaskan tentang Service Registry

Service Registry adalah komponen dalam arsitektur microservices yang bertindak sebagai pusat pendaftaran dan pencarian layanan. Semua layanan dalam sistem akan mendaftarkan diri ke Service Registry, sehingga service discovery (baik client-side maupun server-side) dapat menemukan layanan yang tersedia.

Cara Kerja:

* Ketika sebuah layanan baru dimulai, ia mendaftarkan dirinya ke Service Registry dengan alamat IP dan port-nya.
* Service Registry menyimpan daftar semua layanan aktif dan instance-nya.
* Saat layanan client membutuhkan layanan lain, ia bisa melakukan query ke Service Registry untuk mendapatkan alamat layanan yang tersedia.
* Jika ada instance layanan yang mati atau tidak merespons, Service Registry akan menghapusnya dari daftar.

Contoh Implementasi Service Registry:

* Netflix Eureka
* HashiCorp Consul
* Apache ZooKeeper
* Etcd (dalam Kubernetes)

Service Registry sangat penting dalam arsitektur microservices karena membantu dalam load balancing, failover handling, dan service discovery.