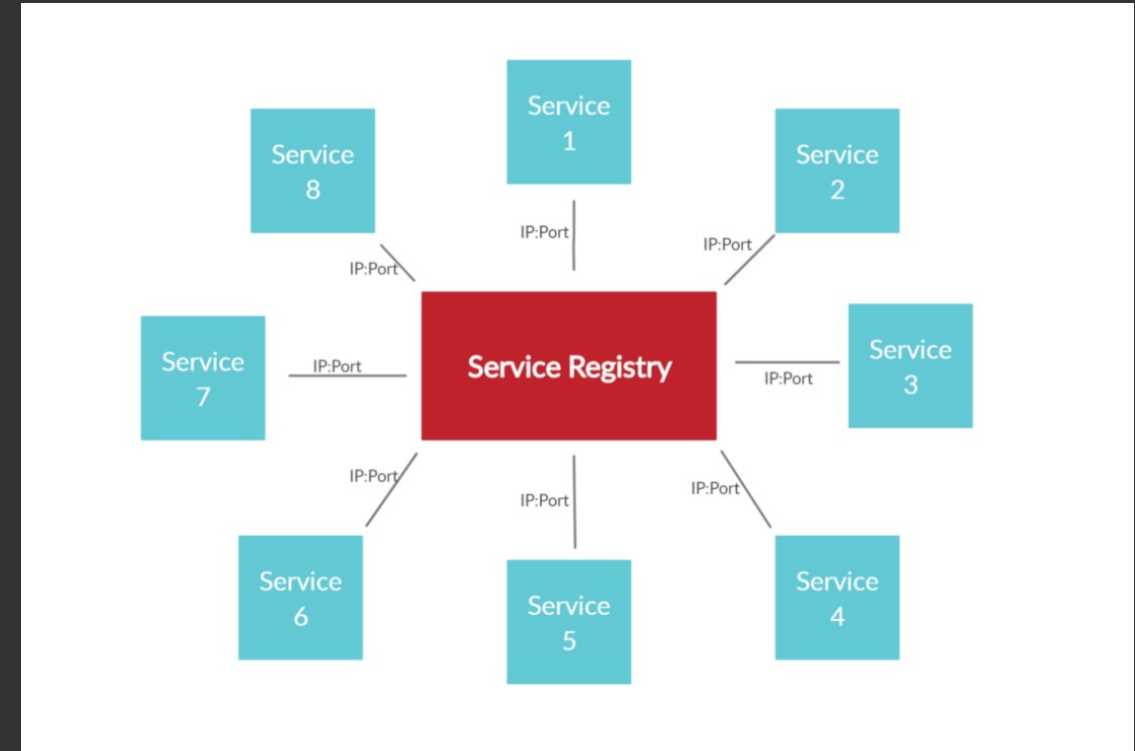
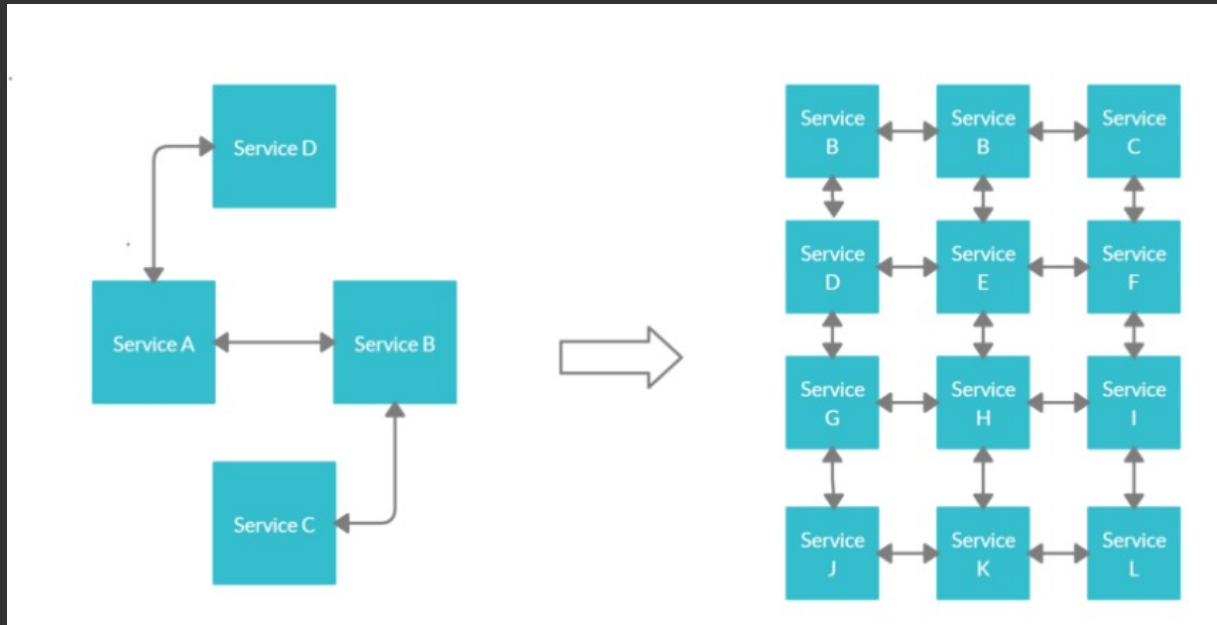




Microservice

Service Registration with Eureka

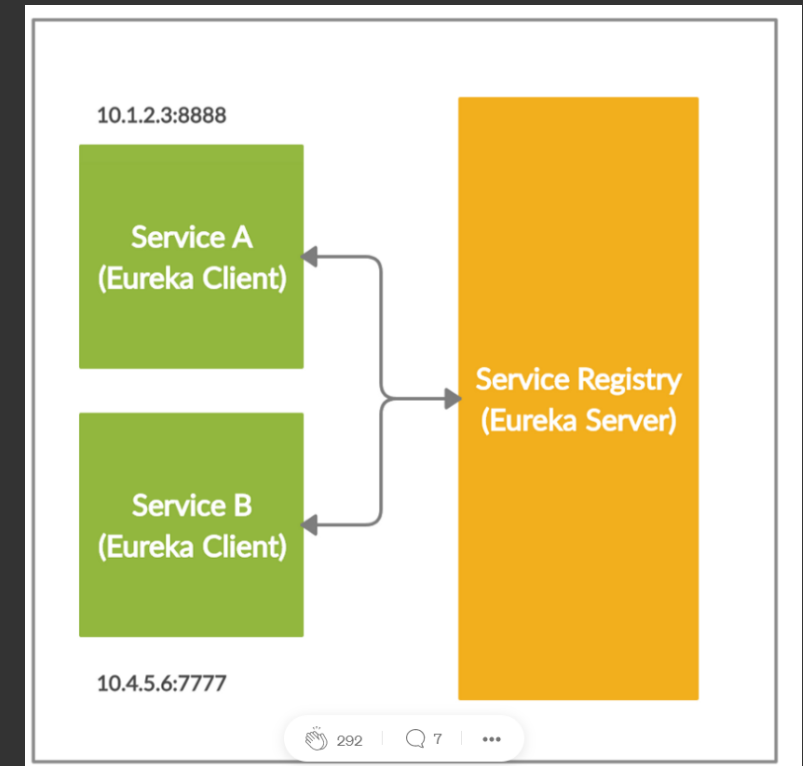


What is Service Discovery

In a microservices architecture, each microservice is a standalone application with specific business functionality. Since these microservices need to communicate with each other to function as a **complete application**, they need to know each other's network locations. Service Discovery comes into play here, maintaining a record of these services' locations, helping them find each other, and enabling communication.

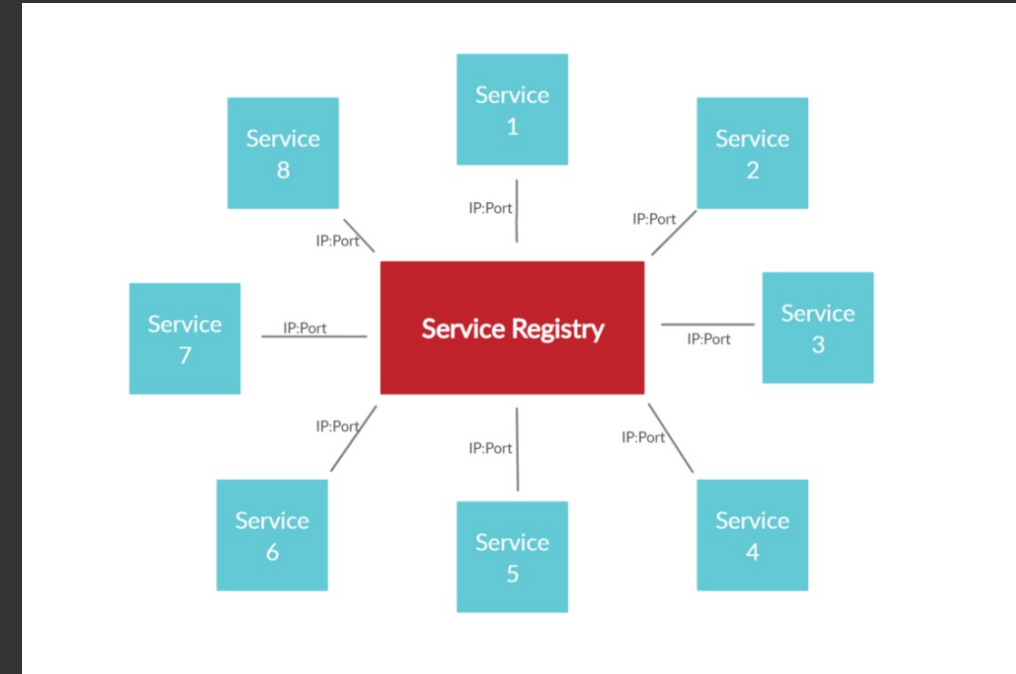
Spring Cloud Eureka

Eureka is a REST based service which is primarily used for acquiring information about services that you would want to communicate with. This REST service is also known as **Eureka Server**. The Services that register in Eureka Server to obtain information about each other are called **Eureka Clients**.



How Eureka Works

- **Service Registration:** A service registers itself with the Eureka Server upon startup.
- **Heartbeat:** The service sends heartbeats periodically to renew its lease with the Eureka Server.
- **Service Discovery:** Other services can query Eureka to discover the location (IP and port) of the registered service.
- **Health Check:** Eureka performs health checks to ensure that registered services are still healthy.
- **Eviction:** If a service stops sending heartbeats and its lease expires, the Eureka Server evicts it from the registry.



Configuring the Eureka Server

```
spring:
  application:
    name: eureka-server
server:
  port: 8761
eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
```

Configuring the Eureka Client

```
1 spring:
2   application:
3     name: <your-service-name> # Name of your microservice
4
5 eureka:
6   client:
7     service-url:
8       defaultZone: http://localhost:8761/eureka/ # URL of the Eureka Server where the client should register
9     fetch-registry: true # (usually true for clients)
10    register-with-eureka: true # (usually true for clients)
11
12 instance:
13   prefer-ip-address: true # Use IP instead of hostname for registration (useful for containers)
14   lease-renewal-interval-in-seconds: 10 # (default is 30)
15   lease-expiration-duration-in-seconds: 30 # (default is 90)
16
17 |
```

