

19-2 Client Service

註冊與管理

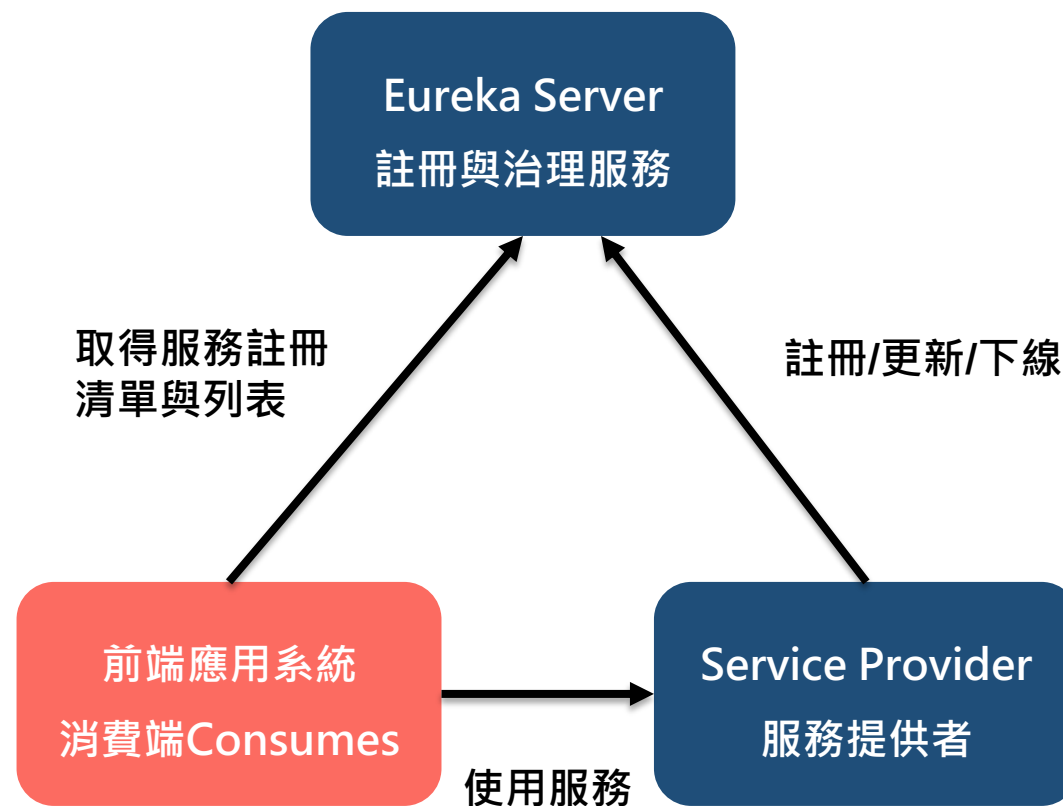


服務發現是基於微服務架構的關鍵原則之一

如果我們嘗試手動配置每個客戶端 服務約定是非常脆弱的

Eureka 是 Netflix 服務用探索服務和客戶端。

可以將服務配置和部署，每個服務都將有關已註冊服務的狀態複製給給其他服務伺服器



專案Groupid ID配置org.springframework.cloud

依賴元件配置

- spring-cloud-starter-eureka

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-dependencies</artifactId>
      <version>${spring-cloud.version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
  <version>1.4.7.RELEASE</version>
</dependency>
```

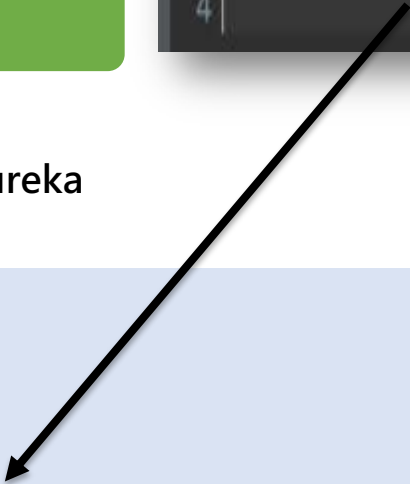
當客戶端向 Eureka 註冊時，它會提供有關自身的元件數據

- 例如主機、端口、健康指標 URL、主頁和其他詳細訊息

Spring Boot前端專案application.properties配置

- spring.application.name=hello-service
- eureka.client.serviceUrl.defaultZone=http://localhost:1111/eureka

```
server.port=1111
eureka.instance.hostname=localhost
eureka.client.register-with-eureka=false
eureka.client.fetch-registry=false
#Eureka Server name
eureka.client.serviceUrl.defaultZone=http://${eureka.instance.hostname}:${server.port}/eureka
```



```
application.properties x eurekaclient/pom.xml
1 server.port=8089
2 spring.application.name=hello-service
3 eureka.client.serviceUrl.defaultZone=http://localhost:1111/eureka
4 |
```

如何向Eureka Spring Boot 類別設定啟動探索前端

Spring Boot 啟動類別中

使用@EnableDiscoveryClient

```
import org.springframework.boot.SpringApplication;

@SpringBootApplication
@EnableDiscoveryClient
@ComponentScan(basePackages= {"com.khh.service"})
public class EurekaurekaclientApplication {

    public static void main(String[] args) {
        SpringApplication.run(EurekaurekaclientApplication.class, args);
    }
}
```

啟動Eureka Server

啟動Eureka Client

```
InstanceInfoReplicator onDemand update allowed rate per min is 4  
Discovery Client initialized at timestamp 1664359096302 with in  
Registering application HELLO-SERVICE with eureka with status UP  
Saw local status change event StatusChangeEvent [timestamp=1664359096302, currentStatus=UP, previousStatus=DOWN]  
DiscoveryClient_HELLO-SERVICE/host.docker.internal:hello-service:8089  
Tomcat started on port(s): 8089 (http) with context path ''  
Updating port to 8089
```

DS Replicas

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
HELLO-SERVICE	n/a (1)	(1)	UP (1) - host.docker.internal:hello-service:8089



總結：19-2 Client Service註冊與管理

設計了前端服務，且進行探索相對Eureka Server進行註冊，接下來我們且來看看消費端如何配合Eureka Server於服務端進行服務呼喚。

