

# Spring Cloud Netflix Eureka 란?

## Netflix Eureka 란?

- 넷플릭스에서 MSA 를 위해 Spring Cloud 에 기부한 오픈 소스이다.

클라우드 환경에서 컨테이너 기반의 배포 시 자동으로 확장 또는 축소 등으로 서비스의 아이피가 동적으로 변경 되는 일이 빈번히 일어난다. 그래서 해당 서비스를 호출할 때 서비스의 위치(아이피 주소와 포트)를 알아낼 수 있는 기능이 필요한데 이를 Service Discovery 라고 한다.

Eureka는 Service Discovery로 각각의 서비스 인스턴스들이 동적으로 확장, 축소 되더라도 인스턴스의 상태를 관리할 수 있는 서비스이다.

## Netflix Eureka 구성 요소

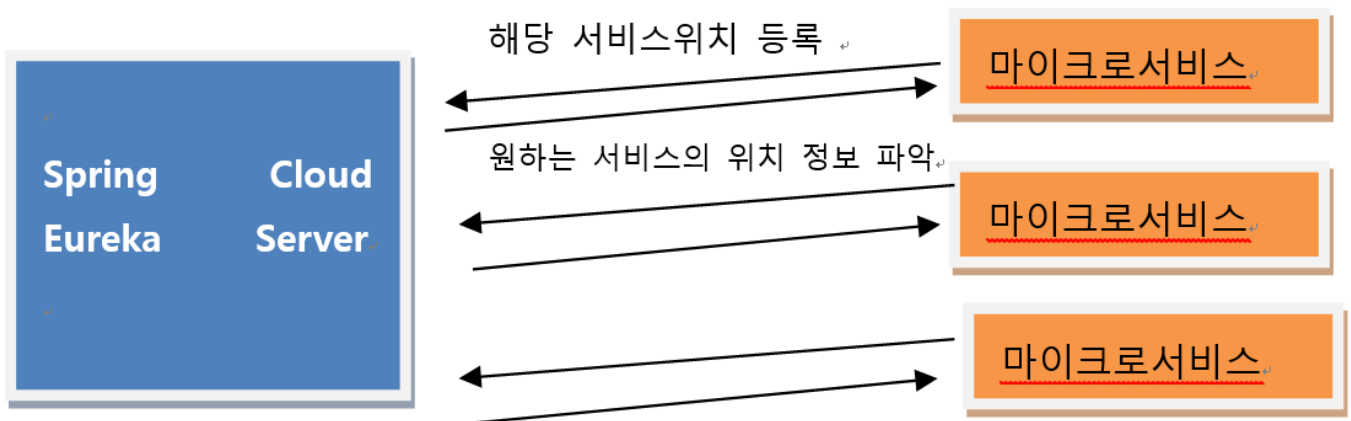
### 1. Service Discovery

- (1) 마이크로 서비스의 위치가 등록된 서버에서 특정 작업을 위한 서버의 위치를 파악하는 작업.
- (2) Spring Cloud Netflix - Eureka Server를 사용

### 2. Service Registry

- (1) 마이크로 서비스가 자신의 위치(IP) 정보를 특정 서버에 등록하는 작업
- (2) Spring Cloud Netflix - Eureka Client를 사용

## Netflix Eureka 동작 흐름



1. Eureka Server 기동
2. 마이크로 서비스 기동 (유레카 클라이언트)
3. 유레카 클라이언트는 서버에 등록이 되며 유레카 서버는 클라이언트에게 30초마다 Ping을 보내면 Health Check(서버의 상태를 주기적으로 확인하여 서버의 정상 작동 여부를 판단하는 과정)을 수행
4. 만약 동작을 하지 않으면 Eureka Server는 해당 Client를 Registry에서 삭제한다.

# Spring Cloud Netflix Eureka Server 만들기

## 1. 프로젝트 생성

Name: [discovery-service](#)

Type: Maven

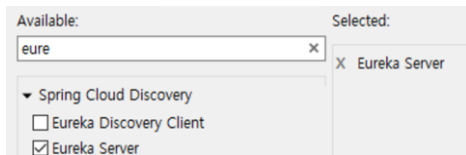
Packaging: Jar

Java Version: 17

Next

Spring Boot Version: 3.1.11

Eureka Server 선택



```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>
</dependency>
```

## 2. main()클래스

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication
@EnableEurekaServer //유레카 서버로 작동하기 위해 사용
public class DiscoveryApplication {

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

}
```

## 3. application.yml

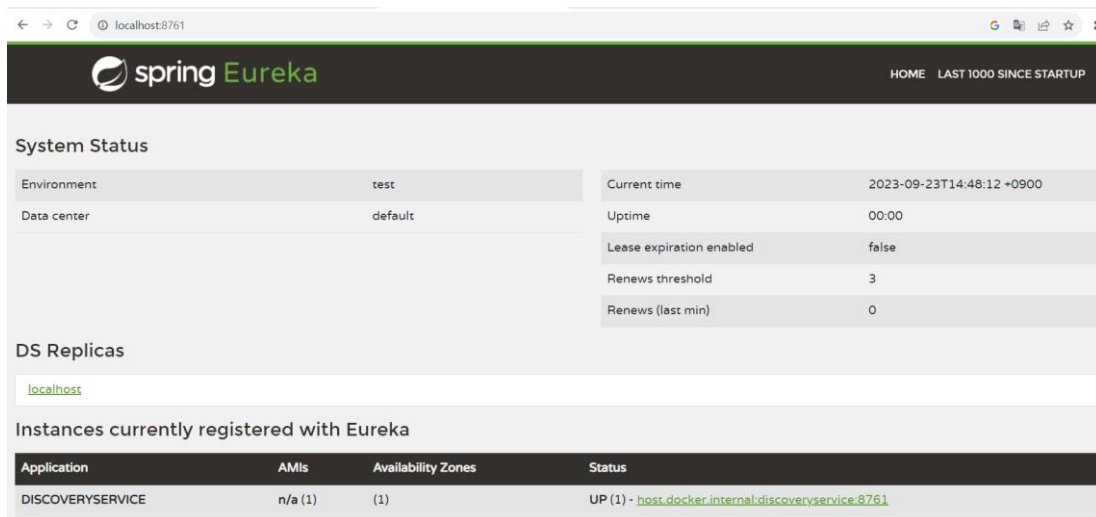
```
server:
  port: 8761

spring:
  application:
    name: discovery-service
```

## 4. Run-As

```
InstanceRegistryImpl : Changing status of 01
InitializerConfiguration : Started Eureka Server
```

## 5. 브라우저 접속



현재 Application DISCOVERYSERVICE가 유레카에 등록됨을 알 수 있다.  
자신은 등록되지 않도록 application.yaml 파일을 수정한다.

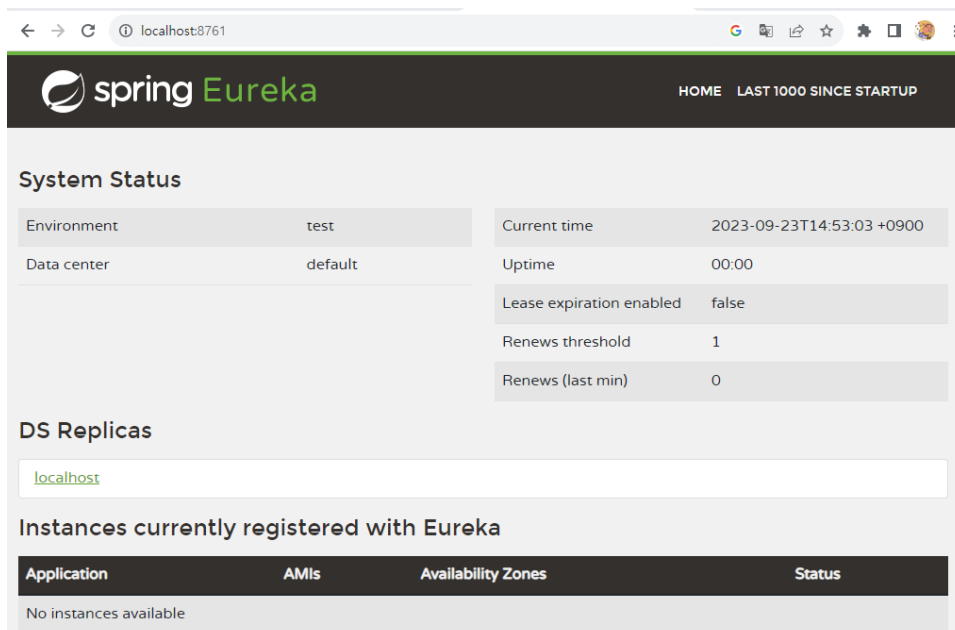
## 6. application.yaml 수정하기

```
server:
  port: 8761

spring:
  application:
    name: discoveryservice

eureka:
  client:
    register-with-eureka: false
    fetch-registry: false
```

## 7. Run - As



등록된 인스턴스가 없다

## Instance Info

Name	Value
ipAddr	192.168.0.23
status	UP

[discoveryervice](#) IP 주소를 확인 할 수 있다.