服务注册中心使用Eureka

高可用注册中心

部署在两台服务器上10.10.11.72和10.10.11.73

http://10.10.11.72和10.10.11.73/eureka/,http://10.10.11.73:9182/eureka/

Eureka服务端

使用注解@EnableEurekaServer启动一个服务注册中心

@EnableEurekaServer

@SpringBootApplication

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

配置文件

eureka:

instance:

hostname: localhost

client:

registerWithEureka: false

fetchRegistry: false

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

基于注册中心的服务发现的核心问题

服务注册：每个服务都需要注册到注册中心

服务发现：每个服务要调用其它服务怎么找到要调用服务的地址

健康检查：某个服务不可用，注册中心如何知晓并剔除

高可用注册中心集群：多个注册中心节点之间数据如何同步

客户端要做的事情

服务注册：主动向Eureka Server端注册。提供自身元数据包括ip、端口等信息

服务续约：每隔30秒向Server发送心跳检测

拉取注册中心列表：从Server获取注册表信息缓存在本地，缓存默认30秒更新一次

Server端要做的事情

服务剔除：客户端连续90秒没有向Server发送服务续约，会将该服务实例从服务注册表中删除

信息同步：高可用的注册中心都是集群，有多个Server。在集群中，每个Eureka Server同时也是Eureka Client

多个Server之间通过复制的方式完成信息的同步

服务注册表 ConcurrentHashMap

处理所有来自eureka客户端的注册表请求。包括注册、更新、状态变化

public abstract class AbstractInstanceRegistry implements InstanceRegistry {

private final ConcurrentHashMap<String, Map<String, Lease<InstanceInfo>>> registry

= new ConcurrentHashMap<String, Map<String, Lease<InstanceInfo>>>();

}

key:就是服务的名称

value：一个服务的多个实例。

value又是一个Map

key：服务实例id

value：InstanceInfo服务实例的具体信息 包括ip、端口、状态

Eureka客户端

@EnableDiscoveryClient

让应用成为Eureka应用，以获得服务发现的能力

添加Eureka客户端依赖

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

配置文件中添加Eureka注册中心地址

eureka:

client:

serviceUrl:

defaultZone: http://localhost:9001/eureka/