Netflix组件之一服务注册中心

1 英文解释

Eureka，英文[juˈriːkə]，翻译我找到了，我发现了。

2 功能

服务注册与发现

3 通讯协议

Client与Server交互使用rest风格的http协议。程序使用HttpClient来发送Http请求。

4 组成

Eureka Client与Eureka Server（注册中心）

5 程序逻辑

微服务启动时，会使用Client向Server注册自己，Server会保存注册信息。

多个Server之间通过Replicate的方式来完成服务注册表的同步，实现Server的高可用。

Client会缓存注册表信息。

续约：Client会默认30s向Server发送心跳包。

续期：Server会60s向Client执行一次服务失效检测功能，还会检查超过默认90s没有发送心跳包的Client，然后进行注销。

Eureka 有一个 Region 和 Zone 的概念，你可以理解为现实中的大区（Region）和机房（Zone）。Eureka Client 在启动时需要指定 Zone，它会优先请求自己 Zone 的 Eureka Server 获取注册列表。同样的，Eureka Server 在启动时也需要指定 Zone，如果没有指定的话，其会默认使用 defaultZone。

详见代码：org.springframework.cloud.netflix.eureka.EurekaClientConfigBean

6 程序

使用@EnableEurekaServer注解来表明当前为Server模式。

使用@EnableEurekaClient注解来表明当前为Client模式。

spring-cloud-netflix-eureka-server-2.0.2.RELEASE.jar

spring-cloud-netflix-eureka-client-2.0.2.RELEASE.jar

eureka.client.register-with-eureka=false

#如果是Server就不需要

eureka.client.fetch-registry=false

#因为是单点Server，所以不需要从别处同步

com.netflix.appinfo.InstanceInfo

实例信息

com.netflix.appinfo.LeaseInfo

租约信息

com.netflix.appinfo.InstanceInfo.InstanceStatus

实例状态

com.netflix.appinfo.InstanceInfo.Builder

建造者模式

com.netflix.appinfo.InstanceInfo.Builder.LazyHolder

使用内部类来实现单例模式的懒加载

com.netflix.appinfo.EurekaInstanceConfig

实例属性接口

int getLeaseRenewalIntervalInSeconds();

org.springframework.cloud.netflix.eureka.InstanceInfoFactory

实例信息工厂，传入[实例属性接口]，返回[实例信息对象]

com.netflix.appinfo.providers.EurekaConfigBasedInstanceInfoProvider

构造实例信息对象，上同

org.springframework.cloud.netflix.eureka.EurekaClientAutoConfiguration

客户端自动配置

com.netflix.eureka.registry.InstanceRegistry

实例注册

com.netflix.discovery.InstanceInfoReplicator

客户端向服务器注册信息

org.springframework.cloud.netflix.eureka.server.EurekaController

是我们访问Eureka的页面的控制器，返回服务器中保存的节点状态信息

Populate填充数据

com.netflix.eureka.EurekaServerContextHolder

静态持有类，持有服务器的状态

com.netflix.discovery.EurekaClientConfig（接口）

Eureka客户端配置，储存着从配置文件中读取的配置属性信息

实现子类2个

org.springframework.cloud.netflix.eureka.EurekaClientConfigBean

com.netflix.discovery.DefaultEurekaClientConfig

一

eureka.client.registry-fetch-interval-seconds

表示eureka client间隔多久去拉取服务注册信息，默认为30秒，对于api-gateway，如果要迅速获取服务注册状态，可以缩小该值，比如5秒

二

com.netflix.appinfo.EurekaInstanceConfig

com.netflix.discovery.EurekaClientConfig

com.netflix.eureka.EurekaServerConfig

eureka.instance.lease-expiration-duration-in-seconds

leaseExpirationDurationInSeconds，表示eureka server至上一次收到client的心跳之后，等待下一次心跳的超时时间，在这个时间内若没收到下一次心跳，则将移除该instance。

默认为90秒

如果该值太大，则很可能将流量转发过去的时候，该instance已经不存活了。

如果该值设置太小了，则instance则很可能因为临时的网络抖动而被摘除掉。

该值至少应该大于leaseRenewalIntervalInSeconds

eureka.instance.lease-renewal-interval-in-seconds

leaseRenewalIntervalInSeconds，表示eureka client发送心跳给server端的频率。如果在leaseExpirationDurationInSeconds后，server端没有收到client的心跳，则将摘除该instance。除此之外，如果该instance实现了HealthCheckCallback，并决定让自己unavailable的话，则该instance也不会接收到流量。

默认30秒

eureka.server.enable-self-preservation

是否开启自我保护模式，默认为true。

默认情况下，如果Eureka Server在一定时间内没有接收到某个微服务实例的心跳，Eureka Server将会注销该实例（默认90秒）。但是当网络分区故障发生时，微服务与Eureka Server之间无法正常通信，以上行为可能变得非常危险了——因为微服务本身其实是健康的，此时本不应该注销这个微服务。

Eureka通过“自我保护模式”来解决这个问题——当Eureka Server节点在短时间内丢失过多客户端时（可能发生了网络分区故障），那么这个节点就会进入自我保护模式。一旦进入该模式，Eureka Server就会保护服务注册表中的信息，不再删除服务注册表中的数据（也就是不会注销任何微服务）。当网络故障恢复后，该Eureka Server节点会自动退出自我保护模式。

综上，自我保护模式是一种应对网络异常的安全保护措施。它的架构哲学是宁可同时保留所有微服务（健康的微服务和不健康的微服务都会保留），也不盲目注销任何健康的微服务。使用自我保护模式，可以让Eureka集群更加的健壮、稳定。

eureka.server.eviction-interval-timer-in-ms

eureka server清理无效节点的时间间隔，默认60000毫秒，即60秒

目前运行关于DiscoveryClient的线程：

DiscoveryClient-0

DiscoveryClient-1

DiscoveryClient-CacheRefreshExecutor-0

DiscoveryClient-HeartbeatExecutor-0

DiscoveryClient-InstanceInfoReplicator-0