Spring Eureka Server的高可用

Eureka Server进行互相注册的方式来实现高可用的部署，所以我们只需要将Eureke Server配置其他可用的serviceUrl就能实现高可用部署

Eureka-Server1 创建application-peer1.properties，作为peer1服务中心的配置，并将serviceUrl指向peer2

spring.application.name=eureka-server1

#delete this if only one eureka server

server.port=1111

eureka.instance.hostname=pee1

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

#eureka.client.serviceUrl.defaultZone=http://localhost:${server.port}/eureka/

eureka.client.serviceUrl.defaultZone=http://pee2:1112/eureka/

Eureka-Server2 创建application-peer2.properties，作为peer2服务中心的配置，并将serviceUrl指向peer1

spring.application.name=eureka-server2

server.port=1112

#delete this if only one eureka server

eureka.instance.hostname=pee2

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

#eureka.client.serviceUrl.defaultZone=http://localhost:${server.port}/eureka/

eureka.client.serviceUrl.defaultZone=http://pee1:1111/eureka/

其他所有服务都注册到pee1 这样就实现了所有注册信息也注册到pee2

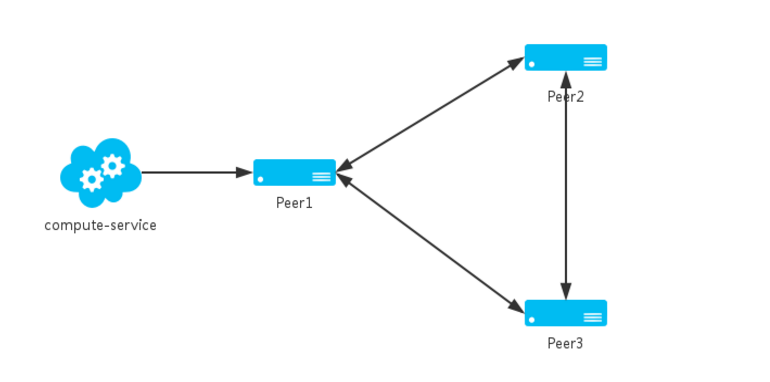
hosts文件中添加对peer1和peer2的转换

127.0.0.1 peer1

127.0.0.1 peer2

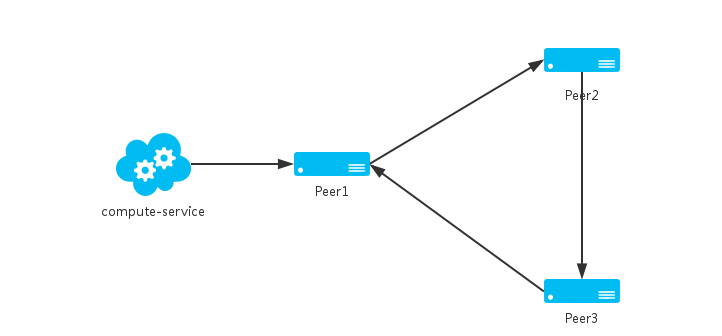
## **场景一**

假设我们有3个注册中心，我们将peer1、peer2、peer3各自都将serviceUrl指向另外两个节点。换言之，peer1、peer2、peer3是两两互相注册的。启动三个服务注册中心，并将compute-service的serviceUrl指向peer1并启动，可以获得如下图所示的集群效果



## **总结：**

* 两两注册的方式可以实现集群中节点完全对等的效果，实现最高可用性集群，任何一台注册中心故障都不会影响服务的注册与发现
* 场景二：依然假设我们有3个注册中心，我们将peer1的serviceUrl指向peer2，peer2的指向peer3，peer3的指向peer1，此时peer1、peer2、peer3通过单向边形成环。分别启动peer1、peer2、peer3，并访问信息页面，我们可以找到下面的规律，peer1成为了peer2的分片节点，peer2成为了peer3的分片节点，peer3则成为了peer1的分片节点，再将compute-service的serviceUrl指向peer1并启动。放别访问peer1、peer2、peer3的信息页面，可以发现compute-service均被peer2和peer3同步过去了，所以单向边也能进行服务的传播与同步。此时，我们断开peer2，可以看到peer3中的compute-service消失了。重新开启peer2并断开peer3，可以看到peer2依然能同步到compute-service。所以我们可以得出结论，Eureka Server的传播与同步是具备方向性的。



我们搭建服务注册中心的高可用集群：

* 两两注册的方式可以实现集群中节点完全对等的效果，实现最高可用性集群，任何一台注册中心故障都不会影响服务的注册与发现
* Eureka Server具备单方面有指向的服务传播与同步机制，在一些对服务发现有限制的情况下，可以利用这样的机制进行服务注册与发现的的单向控制

<http://www.zhongkerd.com/news/content-1392.html>

http://www.cnblogs.com/zhangjianbin/p/6262735.html