# Eureka

## Eureka Environment的配置：

eureka.environment: 字符串

参考文档：

<https://github.com/Netflix/eureka/wiki/Configuring-Eureka>

## Eureka DataCenter的配置

eureka.datacenter: cloud

<https://github.com/Netflix/eureka/wiki/Configuring-Eureka>

这边说：配置-Deureka.datacenter=cloud，这样eureka将会知道是在AWS云上

## Eureka注册服务慢的问题如何解决？

eureka.instance.leaseRenewalIntervalInSeconds

参考文档：

<http://cloud.spring.io/spring-cloud-static/Camden.SR1/#_why_is_it_so_slow_to_register_a_service>

原文：

Why is it so Slow to Register a Service?

Being an instance also involves a periodic heartbeat to the registry (via the client’s serviceUrl) with default duration 30 seconds. A service is not available for discovery by clients until the instance, the server and the client all have the same metadata in their local cache (so it could take 3 heartbeats). You can change the period using eureka.instance.leaseRenewalIntervalInSeconds and this will speed up the process of getting clients connected to other services. In production it’s probably better to stick with the default because there are some computations internally in the server that make assumptions about the lease renewal period.

翻译：

作为实例还涉及到与注册中心的周期性心跳，默认持续时间为30秒（通过serviceUrl）。在实例、服务器、客户端都在本地缓存中具有相同的元数据之前，服务不可用于客户端发现（所以可能需要3次心跳）。你可以使用eureka.instance.leaseRenewalIntervalInSeconds 配置，这将加快客户端连接到其他服务的过程。在生产中，最好坚持使用默认值，因为在服务器内部有一些计算，他们对续约做出假设。

## 如何解决Eureka Server不踢出已关停的节点的问题？

server端:

eureka.server.enable-self-preservation （设为false，关闭自我保护主要）

eureka.server.eviction-interval-timer-in-ms 清理间隔（单位毫秒，默认是60\*1000）

client端：

eureka.client.healthcheck.enabled = true 开启健康检查（需要spring-boot-starter-actuator依赖）

eureka.instance.lease-renewal-interval-in-seconds =10 租期更新时间间隔（默认30秒）

eureka.instance.lease-expiration-duration-in-seconds =30 租期到期时间（默认90秒）

示例：

服务器端配置：

eureka:

    server:

        enableSelfPreservation: false

        evictionIntervalTimerInMs: 4000

客户端配置：

eureka:

    instance:

        leaseRenewalIntervalInSeconds: 10

        leaseExpirationDurationInSeconds: 30

**注意**：

更改Eureka更新频率将打破服务器的自我保护功能

<https://github.com/spring-cloud/spring-cloud-netflix/issues/373>

## Eureka开启自我保护的提示

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

## Eureka配置instanceId显示IP

eureka:

client:

serviceUrl:

defaultZone: http://localhost:8761/eureka/

instance:

preferIpAddress: true

**instance-id: ${spring.cloud.client.ipAddress}:${server.port}**

## Eureka配置最佳实践总结

<https://github.com/spring-cloud/spring-cloud-netflix/issues/203>