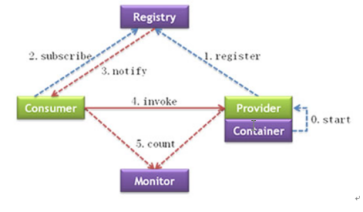
1. SOA 面向服务架构 ， 是分布式架构项目的解决方案 将服务拆分放入不同的节点进行调用。

注意 ： rpc 也能调用远程服务 ，但是当服务越来越多 rpc远程调用需要维护大量的 ip 地址 和 端口号 。 这个时候需要增加一个调度中心 来解决服务调用的 便利 ，这就是SOA 。 dubbo

是SOA的实现 ， 封装ZK 管理数据 ， rpc 实现远程调用

1. Dubbo 分布式服务框架。分布式项目架构的解决方案 ，致力于 提高项目性能 和 rpc 远程调用 服务方案，以及soa 服务治理方案
2. Dubbo 原理



角色 ：

* 1. Provider 服务的 提供方
  2. Container 服务运行的 容器
  3. Consumer 服务的调用方
  4. Registry 注册中心 ： rpc项目调用服务就要写大量的ip和端口号，不利于维护 。 dubbo会讲提供的服务发布到的注册中心，注册中心会维护这些ip 地址 和 端口号 。消费者、提供者只在启动容器的时候 和 注册中心交互
  5. Monitor 监控中心 ：统计服务的调用时间 ，次数等信息

调用顺序关系

1. 启动contanier容器运行服务
2. 将服务对外发布， 信息发布到注册中心
3. 消费者在启动时，需要向注册中心订阅自己的服务来建立关系
4. 注册中心 返回服务提供者的 地址列表给 消费者 ，如果地址有变更 ，注册中心 将基于长连接（一直保持联系） 讲变更信息推送给消费者
5. 然后消费者 在地址列表 中 基于负载均衡算法 ，选一台提供服务者进行调用 ，如果调用失败 则会选择另一台进行调用
6. 监控中心 负责记录 ，调用的时间 次数
7. Dubbo 有 很好的 健壮性

（1）： 注册中心和监控中心全部宕机，不影响已运行的提供者和消费者，消 费者在本地缓存了提供者列表

（2）注册中心（zk）对等集群，任意一台宕掉后，将自动切换到另一台

1. 可视化的Dubbo管理端工具

Service :注册的服务

Consumers ：消费的次数 （客户端启动订阅服务 ，但是并没有消费 只有消费了 才会有值）

1. 怎么用

I、 Dubbo头文件

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:dubbo=*"http://code.alibabatech.com/schema/dubbo"*

xmlns=*"http://www.springframework.org/schema/beans"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://code.alibabatech.com/schema/dubbo [http://code.alibabatech.com/schema/dubbo/dubbo.xsd"](http://code.alibabatech.com/schema/dubbo/dubbo.xsd\">)*[>](http://code.alibabatech.com/schema/dubbo/dubbo.xsd\">)

</beans>

II、服务方 ： 当使用dubbo注解时 ，就必须用dubbo的扫描包配置 ， spring的扫描配置 无需再配

III、消费方 ： 当controller使用dubbo注解 就必须用dubbo的扫描包配置 ， springMVC 就无需再配置 controller的直接