# 面试总结

一、springCloud全家桶成员？

1. Spring Cloud Config：配置管理开发工具包，可以让你把配置放到远程服务器，目前支持本地存储、Git以及Subversion。

2.Spring Cloud Bus：事件、消息总线，用于在集群（例如，配置变化事件）中传播状态变化，可与Spring Cloud Config联合实现热部署。

3.Spring Cloud Netflix（net fa lai ke s）：针对多种Netflix组件提供的开发工具包，其中包括Eureka、Hystrix、Zuul、Archaius等。

4.Netflix Eureka：云端负载均衡，一个基于 REST 的服务，用于定位服务，以实现云端的负载均衡和中间层服务器的故障转移。

5.Netflix Hystrix(ha si chui ke si)：容错管理工具，旨在通过控制服务和第三方库的节点,从而对延迟和故障提供更强大的容错能力。

6.Netflix Zuul(zu ao)：边缘服务工具，是提供动态路由，监控，弹性，安全等的边缘服务。

7.Netflix Archaius(a kai ke)：配置管理API，包含一系列配置管理API，提供动态类型化属性、线程安全配置操作、轮询框架、回调机制等功能。

8.Spring Cloud for Cloud Foundry(fang zhui)：通过Oauth2协议绑定服务到CloudFoundry，CloudFoundry是VMware推出的开源PaaS云平台。

9.Spring Cloud Sleuth( su si)：日志收集工具包，封装了Dapper,Zipkin和HTrace操作。

10.Spring Cloud Data Flow：大数据操作工具，通过命令行方式操作数据流。\

11.Spring Cloud Security：安全工具包，为你的应用程序添加安全控制，主要是指OAuth2。

12.Spring Cloud Consul：封装了Consul操作，consul是一个服务发现与配置工具，与Docker容器可以无缝集成。

13.Spring Cloud Zookeeper：操作Zookeeper的工具包，用于使用zookeeper方式的服务注册和发现。

14.Spring Cloud Stream：数据流操作开发包，封装了与Redis,Rabbit、Kafka等发送接收消息。

15.Spring Cloud CLI：基于 Spring Boot CLI，可以让你以命令行方式快速建立云组件。

## 什么是熔断服务？

服务熔断一般是指软件系统中，由于某些原因使得服务出现了过载现象，为防止造成整个系统故障，从而采用的一种保护措施，所以很多地方把熔断亦称为过载保护。

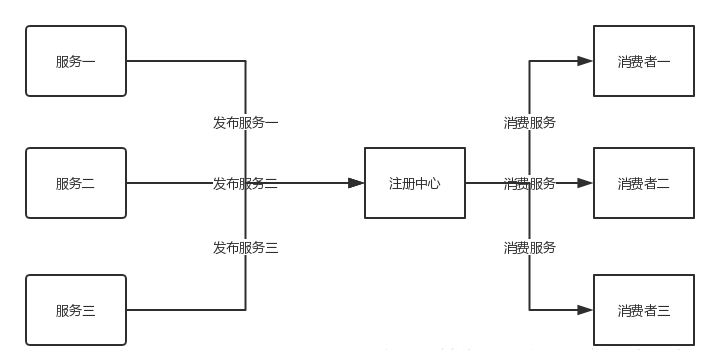
## springcloud如何实现服务的注册和发现？

  服务在发布时 指定对应的服务名（服务名包括了IP地址和端口） 将服务注册到注册中心（eureka或者zookeeper）这一过程是springcloud自动实现 只需要在main方法添加@EnableDisscoveryClient  同一个服务修改端口就可以启动多个实例

        调用方法：传递服务名称通过注册中心获取所有的可用实例 通过负载均衡策略调用（ribbon和feign）对应的服务

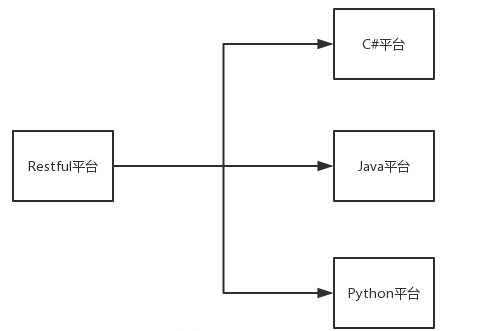
## 四、那么什么是注册中心呢？

注册中心，是服务的提供者发布自己服务的地方，传统手写的restful接口提供的服务会涉及到服务发布者的端口IP等等，通过注册中心可以直接调用已经发布在上面的服务，只需要通过服务名即可。

从图中可以看出，服务提供者将自己的服务发布到注册中心，消费者从注册中心取出服务，然后消费，这样生产者只需要关注自身的业务，将自己的服务发布到注册中心就行了，消费者无需关注生产者的内容，只需要从注册中心取出服务就行了，这样将生产者和消费者直接的耦合给断绝了。

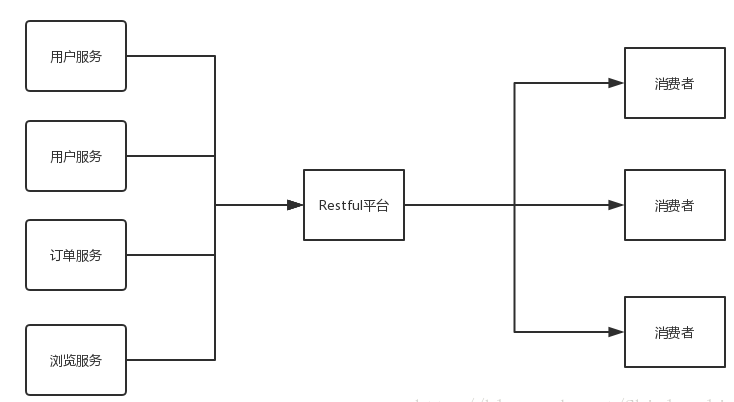
## 五、为什么使用Restful?

首先springcloud对比dubbo，最大的特点之一就是使用Restful的模式进行交互，dubbo是基于RPC进行通信的，而Restful是基于Http协议进行的，从协议的角度上来说Http和RPC都是基于TCP进行研发的协议，Http是最广泛的，不仅支持浏览器还支持各种APP通信，这么来说吧Http就是大家都在用的协议，而RPC是针对某一个平台某一个环节针对性开发的自定义协议，Http由于大众化，所以本身协议会有点笨重，解析起来自然也比RPC要慢，这也是RPC的优势之一，但是Http具有良好的跨平台性质，如下图：



## SpringCloud服务的横向拓展？

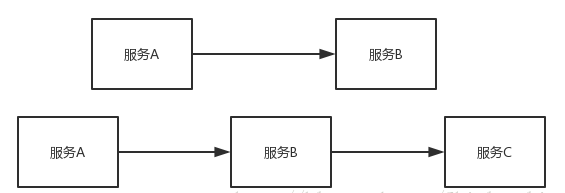
使用SpringCloud的另一个目的就是便于服务的横向拓展，大家都知道，当某一个服务由于访问压力变成瓶颈的时候，我们常常希望这个服务能进行负载均衡，分摊压力，以便于更好的像外界提供服务



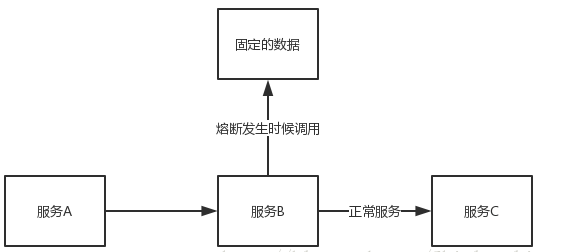
我们可以在图片中看出，用户服务是具有两个实例的，但是消费者并不知道用户服务是具有两个实例，消费者只知道，当前有用户服务对外提供服务，所以消费者只需要知道服务名就行了，不用在意是哪个服务实例对其提供服务，这样也是进一步封装生产者对外提供服务，同时做了负载均衡，这样加入用户服务突然增加业务量，那么我们只需要再运行多个用户服务的实例即可。

## SpringCloud熔断机制？

在开始熔断机制原理讲解之前,先理解一下什么是服务依赖。



从图片上来看，第一行服务A是调用服务B，第二行服务A调用服务B，服务B调用服务C，这样就有一个服务链，A->B->C，假设现在由于网络动荡或者服务器奔溃的原因，服务C挂掉了，然而服务A的访问仍然很大，这样服务A继续请求服务B，而服务B由于无法调用服务C一直在等待，最后导致请求过多，服务B也被压垮了。



当C服务停止的时候，B自动调用写死的数据进行回复，从而避免因为请求过多导致A服务奔溃的情况。

## 什么是微服务？

通常而言，微服务架构是一种架构模式或者说是一种架构风格，它提倡将单一应用程序划分为一组小的服务，每个服务运行在其独立的自己的进程中，服务之间互相协调、互相配合，为用户提供最终价格。服务之间采用轻量级的通信机制互相沟通（通常是基于HTTP的RESTful API）。每个服务都围绕着具体业务进行构建，并且能够被独立地部署到生产环境、类生产环境等。另外应尽量避免统一的、集中式的服务管理机制，对具体的一个服务而言，应柑橘业务上下文，选择合适的语言、工具对其进行构建，可以有一个非常轻量级的集中式管理来协调这些服务，可以使用不同的语言来编写服务，也可以使用不同的数据存储。

## 九、微服务之间是如何独立通讯的？

1. 同步：RPC，REST等

2. 异步：消息队列。要考虑消息可靠传输、高性能，以及编程模型的变化等。

## 十、springCloud与Dubbo有哪些区别？

## 微服务主要的优势？

1、降低复杂度

2、可独立部署

3、容错

4、扩展

## 微服务的优缺点分别是什么？

优点：

  每个服务足够内聚，足够小，代码容易理解这样能聚焦一个指定的业务功能或业务需求

开发简单、开发效率提高，一个服务可能就是专一的只干一件事。

微服务能够被小团队独立开发，这个团队可以使2到5人的开发人员组成。

微服务是松耦合的，是有功能意义的服务，无论实在开发阶段或部署阶段都是独立的。

微服务能使用不同的语言开发。

微服务只是业务逻辑的代码，不会和HTML,CSS或其他界面组件混合。

每个微服务都有自己的存储能力，可以有自己的数据库。也可以有统一的数据库。

缺点：

  开发人员要处理分布式系统的复杂性

多服务运维难度，随着服务的增加，运维的压力也在增加

系统部署依赖

服务间通信成本

数据一致性

系统集成测试

性能监控。。。。。。