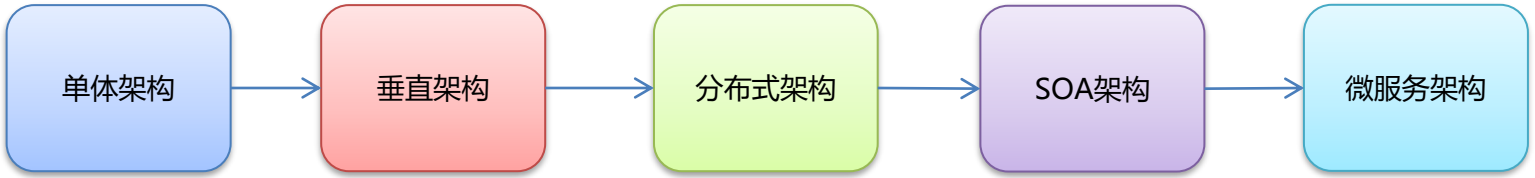
Spring Cloud

目录 Contents

初识 Spring Cloud

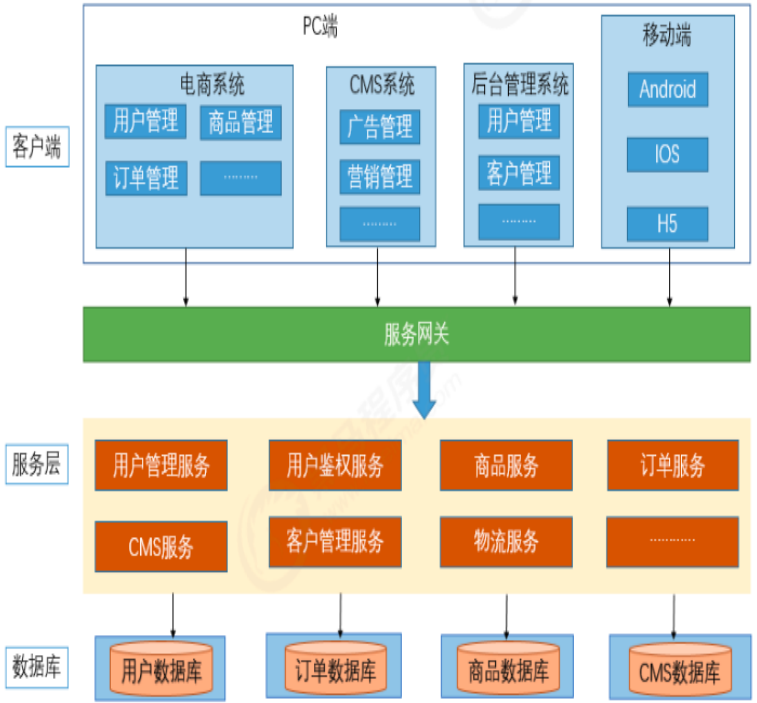
- 微服务架构
- 走进 Spring Cloud
- Spring Cloud 与 Dubbo 对比

微服务架构



微服务架构

- "微服务”一词源于 Martin Fowler的名为 Microservices 的博文,可以在他的官方博客上找到 http://martinfowler.com/articles/microservices.html
- 微服务是系统架构上的一种设计风格,它的主旨是将一个原本独立的系统拆分成多个小型服务,这些小型服务都在各自独立的进程中运行,服务之间一般通过 HTTP 的 RESTfuL API 进行通信协作。
- 被拆分成的每一个小型服务都围绕着系统中的某一项或些耦合度较高的业务功能进行构建,并且每个服务都维护着白身的数据存储、业务开发自动化测试案例以及独立部署机制。
- 由于有了轻量级的通信协作基础,所以这些微服务可以使用不同的语言来编写。



初识 Spring Cloud

- 微服务架构
- 走进 Spring Cloud
- Spring Cloud 与 Dubbo 对比

初识 Spring Cloud

走进 Spring Cloud

- Spring Cloud 是一系列框架的有序集合。
- Spring Cloud 并没有重复制造轮子，它只是将目前各家公司开发的比较成熟、经得起实际考验的服务框架组合起来。
- 通过 Spring Boot 风格进行再封装屏蔽掉了复杂的配置和实现原理，最终给开发者留出了一套简单易懂、易部署和易维护的分布式系统开发工具包。
- 它利用Spring Boot的开发便利性巧妙地简化了分布式系统基础设施的开发，如服务发现注册、配置中心、消息总线、负载均衡、 断路器、数据监控等，都可以用Spring Boot的开发风格做到一键启动和部署。
- Spring Cloud项目官方网址：https://spring.io/projects/spring-cloud

走进 Spring Cloud

- Spring Cloud 版本命名方式采用了伦敦地铁站的名称，同时根据字母表的顺序来对应版本时间顺序，比如：最早的Release版本：Angel，第二个Release版本：Brixton，然后是Camden、Dalston、Edgware，Finchley，Greenwich，Hoxton。
- 目前最新的是Hoxton版本。

Release Train	Boot Version
Hoxton	2.2.x
Greenwich	2.1.x
Finchley	2.0.x
Edgware	1.5.x
Dalston	1.5.x

初识 Spring Cloud

- 微服务架构
- 走进 Spring Cloud
- Spring Cloud 与 Dubbo 对比

Spring Cloud 与 Dubbo 对比

	Dubbo	Spring Cloud
服务注册中心	Zookeeper	Spring Cloud Netflix Eureka
服务调用方式	RPC	REST API
服务监控	Dubbo-monitor	Spring Boot Admin
断路器	不完善	Spring Cloud Netflix Hystrix
服务网关	无	Spring Cloud **Gateway**
分布式配置	无	Spring Cloud Config
服务跟踪	无	Spring Cloud Sleuth
消息总线	无	Spring Cloud Bus
数据流	无	Spring Cloud Stream
批量任务	无	Spring Cloud Task

- Spring Cloud 与 Dubbo 都是实现微服务有效的工具。
- Dubbo 只是实现了服务治理，而 Spring Cloud 子项目分别覆盖了微服务架构下的众多部件。
- Dubbo 使用 RPC 通讯协议，Spring Cloud 使用 RESTful 完成通信，Dubbo 效率略高于 Spring Cloud。

小结

- 微服务就是将项目的各个模块拆分为可独立运行、部署、测试的架构设计风格。
- Spring 公司将其他公司中微服务架构常用的组件整合起来，并使用 SpringBoot 简化其开发、配置。称为 Spring Cloud
- Spring Cloud 与 Dubbo都是实现微服务有效的工具。Dubbo 性能更好，而 Spring Cloud 功能更全面。

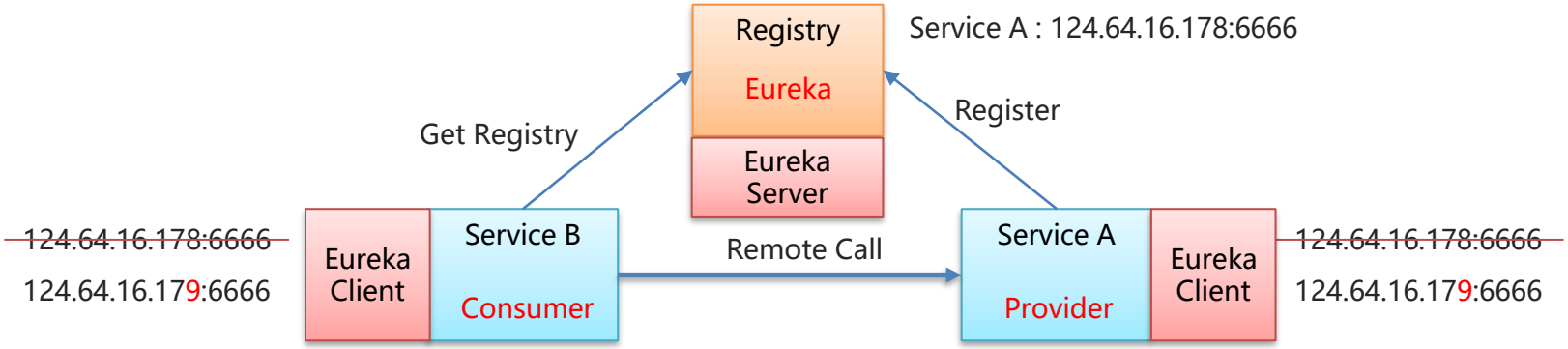
目录 Contents

Spring Cloud 服务治理

- Eureka
- Consul
- Nacos

Eureka

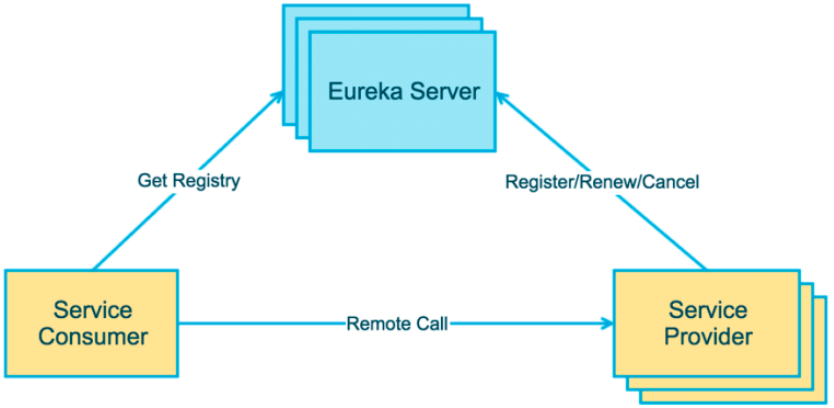
- Eureka 是 Netflix 公司开源的一个服务注册与发现的组件 。
- Eureka 和其他 Netflix 公司的服务组件（例如负载均衡、熔断器、网关等） 一起，被 Spring Cloud 社区整合为 Spring-Cloud-Netflix 模块。
- Eureka 包含两个组件：Eureka Server (注册中心) 和 Eureka Client (服务提供者、服务消费者)。



Spring Cloud 服务治理

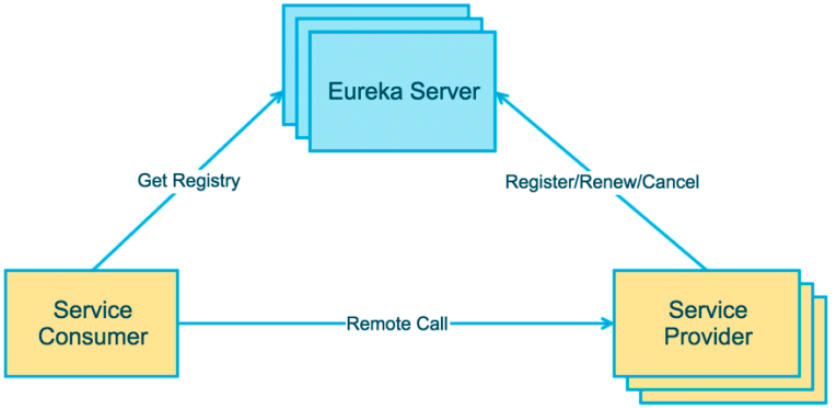
Eureka

1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



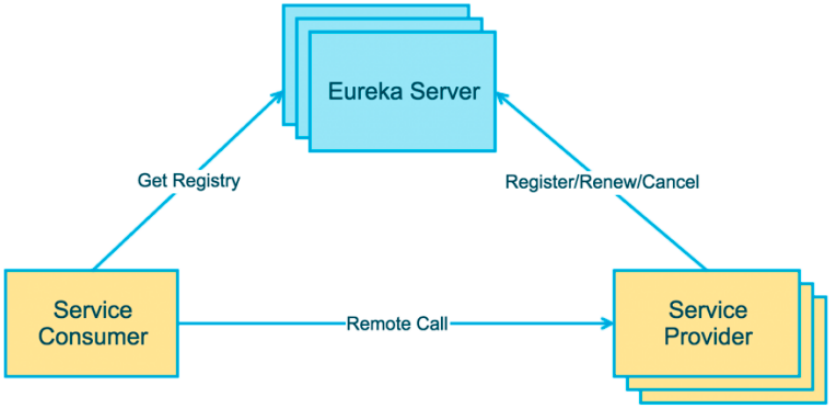
Eureka – 搭建服务

1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



Eureka – RestTemplate

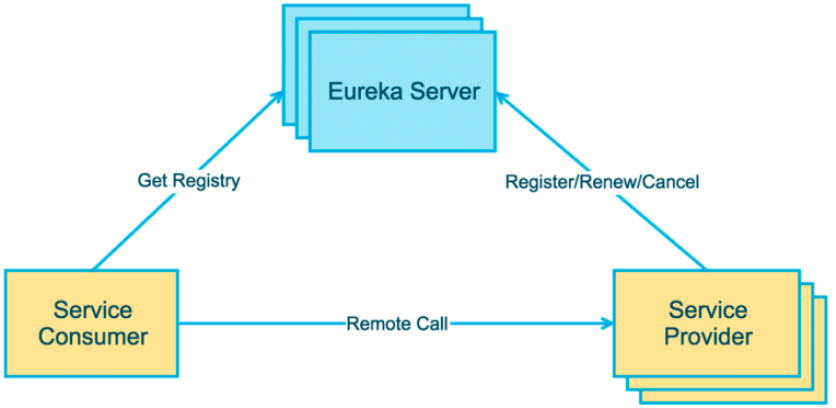
1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



- Spring提供的一种简单便捷的模板类，用于在 java 代码里访问 restful 服务。
- 其功能与 HttpClient 类似，但是 RestTemplate 实现更优雅，使用更方便。

Eureka – Eureka Server

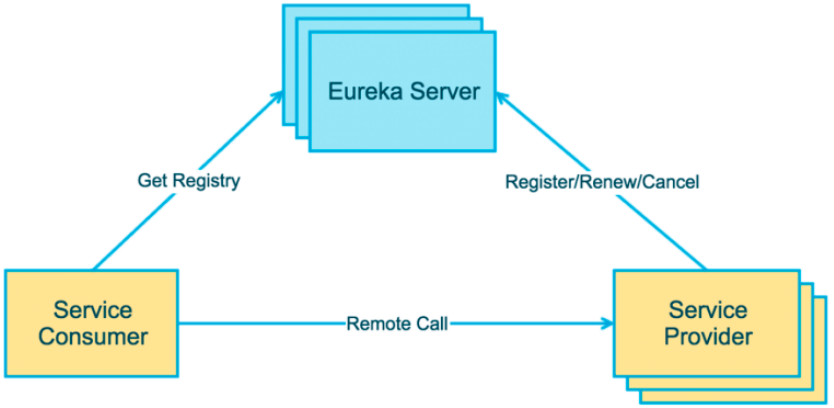
1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



① 创建 eureka-server 模块
② 引入 SpringCloud 和 euraka-server 相关依赖
③ 完成 Eureka Server 相关配置
④ 启动该模块

Eureka – Eureka Client

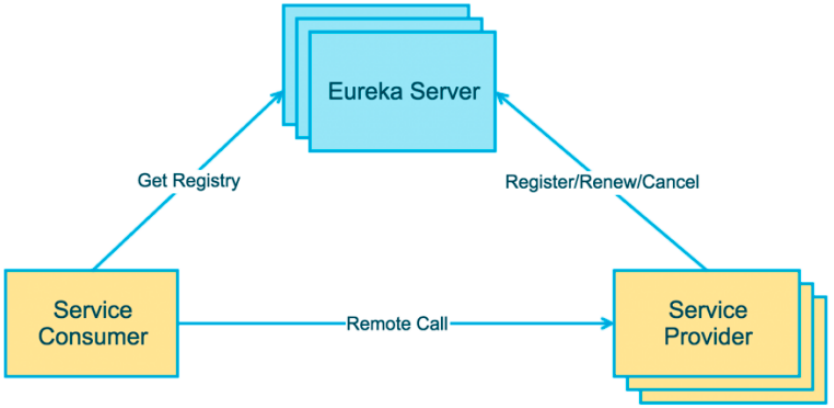
1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



① 引 eureka-client 相关依赖
② 完成 eureka client 相关配置
③ 启动 测试

Eureka – 远程调用

1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 搭建 Eureka Server 服务。
4. 改造 Provider 和 Consumer 称为 Eureka Client。
5. Consumer 服务 通过从 Eureka Server 中抓取 Provider 地址 完成 远程调用



Eureka – 相关配置及特性

eureka 一共有4部分 配置

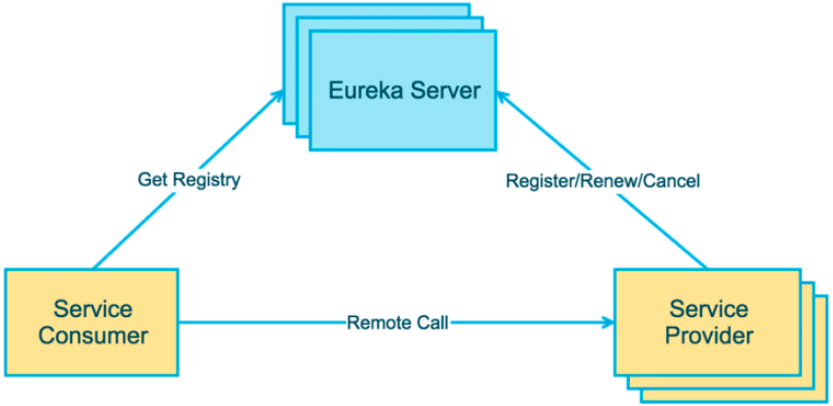
1. server : eureka 的服务端配置
2. client : eureka 的客户端配置
3. instance : eureka 的实例配置
4. dashboard : eureka 的web控制台配置

Eureka – 相关配置及特性 - instance

```yaml
eureka:
  instance:
    hostname: localhost # 主机名
    prefer-ip-address: # 是否将自己的ip注册到eureka中，默认false 注册 主机名
    ip-address: # 设置当前实例ip
    instance-id:  # 修改instance-id显示
    lease-renewal-interval-in-seconds: 30 # 每一次eureka client 向 eureka server发送心跳的时间间隔
    lease-expiration-duration-in-seconds: 90 # 如果90秒内eureka server没有收到eureka client的心跳包，则剔除该服务
```

Eureka – 相关配置及特性 - server

```yaml
eureka:
  server:
    #是否开启自我保护机制，默认true
     enable-self-preservation:
    #清理间隔（单位毫秒，默认是60*1000）
     eviction-interval-timer-in-ms:
```



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.

DS Replicas

```yaml
eureka:
  instance:
    lease-renewal-interval-in-seconds: 30 # 每一次eureka client 向 eureka server发送心跳的时间间隔
    lease-expiration-duration-in-seconds: 90 # 如果90秒内eureka server没有收到eureka client的心跳包，则剔除该服务
```

Eureka – 相关配置及特性 - client

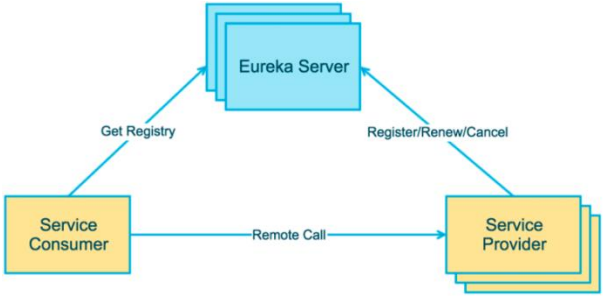
```yaml
eureka:
  client:
    service-url:
        # eureka服务端地址，将来客户端使用该地址和eureka进行通信
        defaultZone:
    register-with-eureka: # 是否将自己的路径 注册到eureka上。
    fetch-registry: # 是否需要从eureka中抓取数据。
```

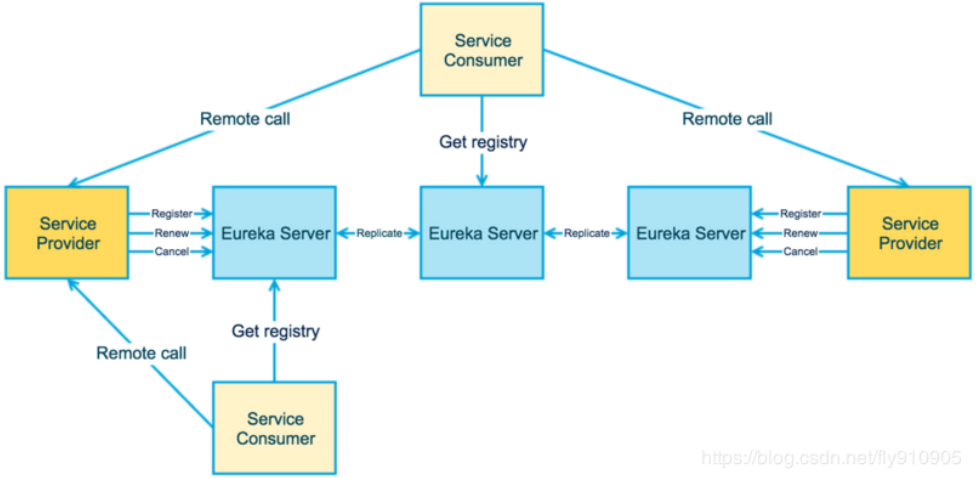
Eureka – 相关配置及特性 - dashboard

```yaml
eureka:
  dashboard:
    enabled: true # 是否启用eureka web控制台
    path: / # 设置eureka web控制台默认访问路径
```

Spring Cloud 服务治理

Eureka – 高可用





1. 准备两个Eureka Server
2. 分别进行配置，相互注册
3. Eureka Client 分别注册到这两个 Eureka Server中

Spring Cloud 服务治理

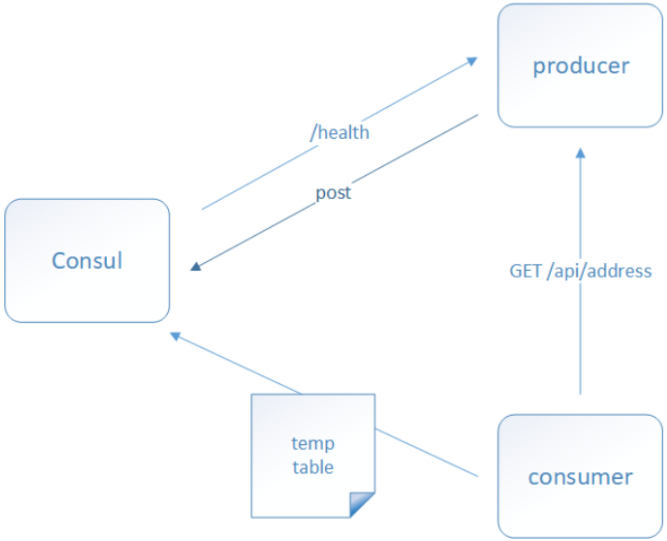
- Eureka
- Consul
- Nacos

Consul

- Consul 是由 HashiCorp 基于 Go 语言开发的，支持多数据中心，分布式高可用的服务发布和注册服务软件。
- 用于实现分布式系统的服务发现与配置。
- 使用起来也较 为简单。具有天然可移植性(支持Linux、windows和Mac OS X)；安装包仅包含一个可执行文件，方便部署 。
- 官网地址： https://www.consul.io

Consul – 快速入门

1. 搭建 Provider 和 Consumer 服务。
2. 使用 RestTemplate 完成远程调用。
3. 将Provider服务注册到Consul中。
4. Consumer 服务 通过从 Consul 中抓取 Provider 地址 完成 远程调用



Spring Cloud 服务治理

- Eureka
- Consul
- Nacos

Nacos

- Nacos（Dynamic Naming and Configuration Service） 是阿里巴巴2018年7月开源的项目。
- 它专注于服务发现和配置管理领域 致力于帮助您发现、配置和管理微服务。Nacos 支持几乎所有主流类型的“服务”的发现、配置和管理。
- 一句话概括就是Nacos = Spring Cloud注册中心 + Spring Cloud配置中心。
- 官网：https://nacos.io/
- 下载地址： https://github.com/alibaba/nacos/releases

Nacos – 快速入门

传智播客旗下高端IT教育品牌