



DevOpsDays

Shanghai

— 2017.8.18-8.19 —

上海龙之梦酒店（长宁区延安西路1116号）

主办单位：



高效运维社区
GreatOPS Community



Best Practice
最佳实践





如何使用Spring Cloud快速构建微服务应用

许进(xujin.org)

About Me

- 目前就职于众安保险中间件框架组
- 先后就职于唯品会，饿了么。
- 网站:<http://xujin.org>
- Spring Cloud 中国社区(springcloud.cn)发起人
- 曾就职于唯品会基础架构部，从事中间件研发工作。
- 曾就职于唯品会金融事业部，参与唯品会金融平台(<https://jinrong.vip.com/>)的设计与开发

目录

- ➔ **1** Spring Cloud概述
- 2** Spring Cloud国内用情况
- 3** 使用Spring Cloud搭建微服务应用
- 4** 展望Spring Cloud未来
- 5** Q&A

Spring Cloud概述

以下简称：SC (Spring Cloud)

Spring Cloud，基于Spring Boot提供了一套微服务解决方案，包括服务注册与发现，配置中心，全链路监控，服务网关，负载均衡，熔断器等组件，除了基于Netflix的开源组件做高度抽象封装之外，还有一些选型中立的开源组件。



<https://github.com/spring-cloud>

为什么要选择Spring Cloud?

	SpringCloud	Dubbo	Motan	Thrift
定位	完整微服务框架	RPC框架	服务框架	Rpc框架
服务注册/发现	是	是	是	否
负载均衡	是	是	是	否
配置服务	是	否	是	否
高容错	是	是	是	否
调用方式	Rest/rpc(未来)	rpc	rpc	rpc
社区活跃度	高	弃坑	一般	一般
学习难度	低	中	低	高
文档丰富度	高	高	一般	一般

目录

1 Spring Cloud概述

➔ **2** Spring Cloud国内使用情况

3 我是一个标题

4 我是一个标题

5 我是一个标题

6 我是一个标题

Spring Cloud在国内使用情况

1. 中国联通子公司

http://fp.baidu.com/feedland/video?entry=bbox_search:bbox_feed&id=144115189637730162&from=timeline&isappinstalled=0

2. 上海米么金服

3. 指点无限（北京）科技有限公司

4. 易保软件 目前正在定制开发中

<http://www.ebaotech.com/cn/>

5. 广州简法网络

6. 深圳睿云智合科技有限公司

持续交付产品基于Spring Cloud研发 <http://www.wise2c.com>

7. 猪八戒网

8. 上海云首科技有限公司

9. 华为

整合netty进来用rpc 包括nexus私服东西 需要注意的是sleuth traceid的传递需要自己写。tps在物理机上能突破20w

10. 东软

11. 南京云帐房网络科技有限公司

12. 四众互联(北京)网络科技有限公司

13. 深圳摩拿技术科技有限公司

14. 广州万表网

15. 视觉中国

16. 上海泰谷信息科技有限公司-买单侠

17. 爱油科技(大连)有限公司

爱油科技基于SpringCloud的微服务实践

18. 广发银行

19. 卖货郎(<http://www.51mhl.com/>)

20. 拍拍贷

21. 甘肃电信

22. 新浪商品部

23. 春秋航空

24. 冰鉴科技

25. 万达网络科技集团-共享商业平台-共享供应链中心

26. 网易乐得技术团队

27. 饿了么某技术团队

28. 高阳捷迅信息科技有限公司-话费中心业务平台-凭证查询及收单系统
数据在统计之中，会一直持续更新，敬请期待！



网易乐得专栏

网易乐得技术团队，
Spring Cloud分享专



广发银行
CHINA GUANGFA BANK



目录

1 Spring Cloud概述

2 Spring Cloud国内使用情况

➔ 3 如何使用Spring Cloud构建微服务

4 展望Spring Cloud与社区

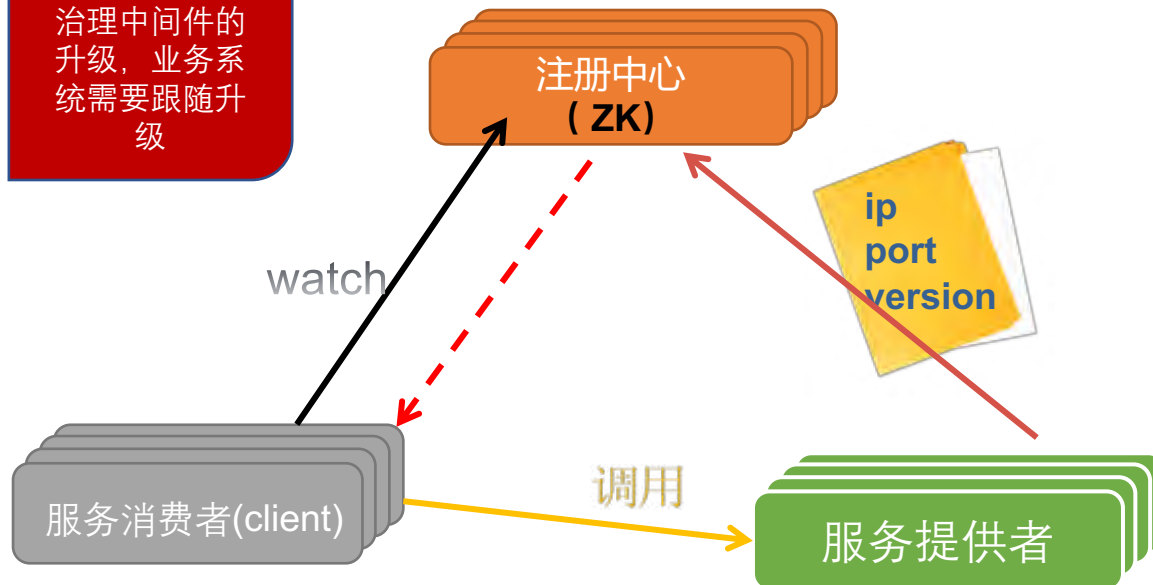
5 Q&A

Spring Cloud的服务注册与发现

由于Spring Cloud的选型中立，特别是spring-cloud-commons项目的高度抽象，可以自由选型，在改变极小的情况下切换组件。比如从Eureka，切换到Consul，ZK，或者Etcd。

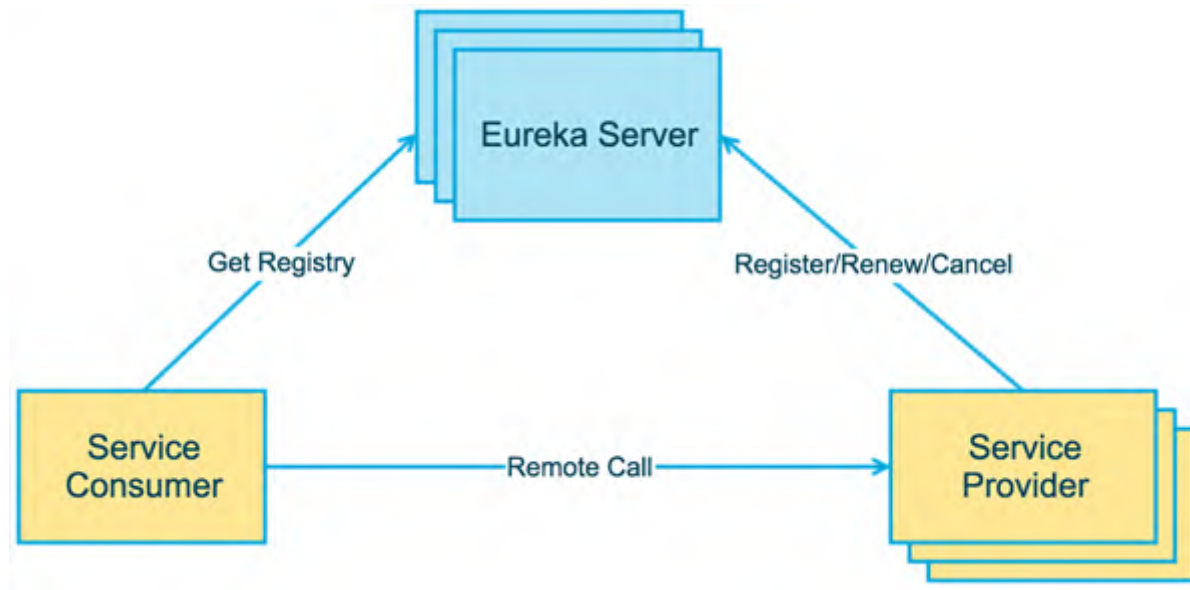
服务注册与发现

缺点:随着服务治理中间件的升级,业务系统需要跟随升级



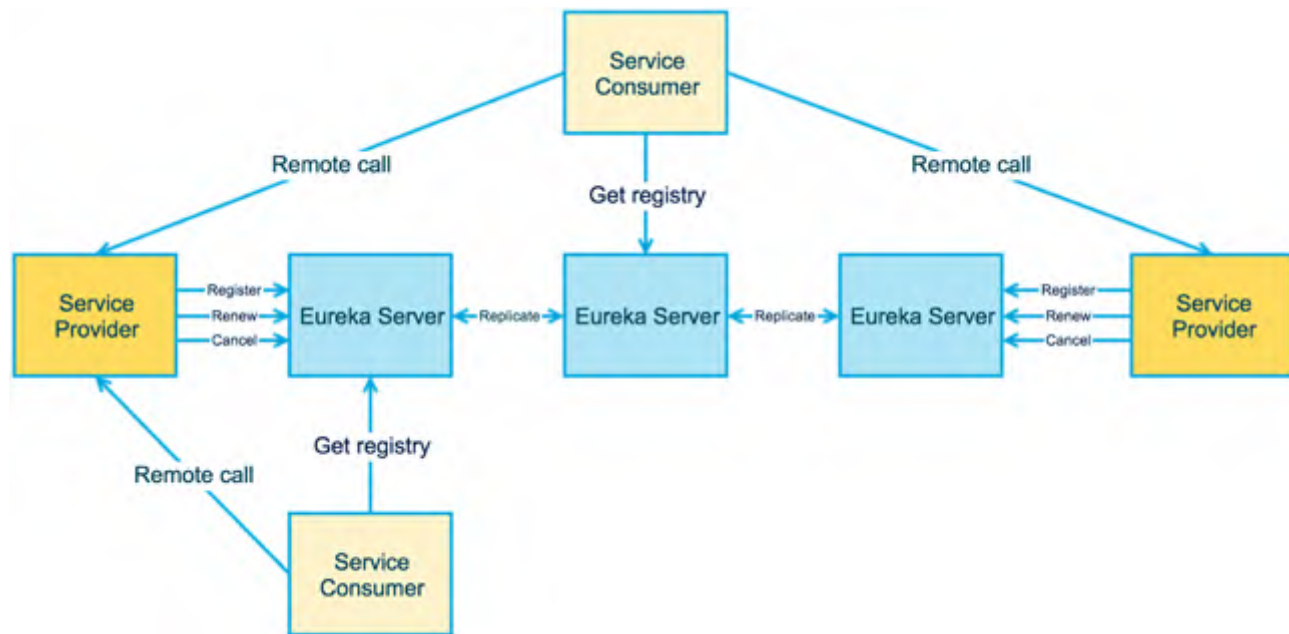
基于客户端的服务治理, 比如Dubbo, Spring Cloud

Eureka基本架构图



<http://xujin.org/sc/sc-netflix-eureka/>

集群模式下的Eureka



<http://xujin.org/sc/sc-netflix-eureka/>

使用Eureka快速搭建注册中心

```

/**
 * Eureka Server
 * @author xujin
 */
@SpringBootApplication
@EnableEurekaServer
public class SpringCloudEurekaServer {

    public static void main(String[] args) {
        SpringApplication.run(SpringCloudEurekaServer.class, args);
    }
}

server:
  port: 8761 # 指定该Eureka实例的端口
eureka:
  client:
    #表示是否将自己注册到Eureka Server上, 默认为true, 当前
    registerWithEureka: false
    #表示是否从Eureka Server获取注册信息, 默认为true。因为
    fetchRegistry: false
    #Eureka Server的访问地址, 服务注册和client获取服务注册
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/

```



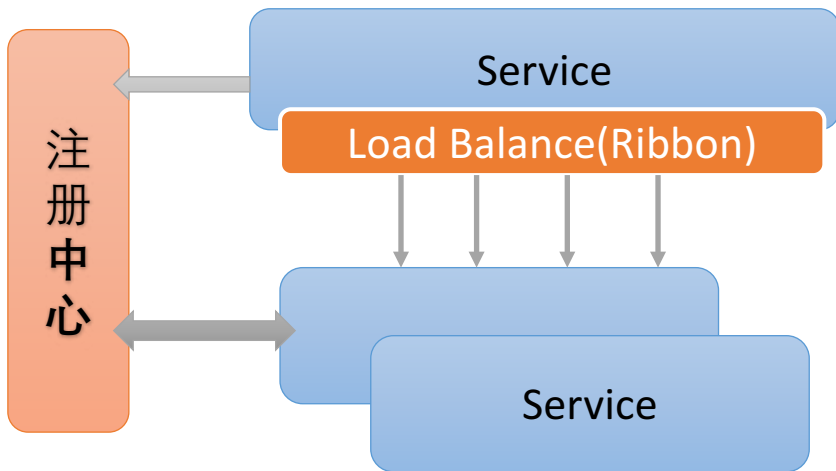
DevOpsDays 2017·上海站

Spring Cloud的Ribbon

LB，即负载均衡(Load Balance)，在微服务或分布式集群中经常用的一种应用。负载均衡简单的说就是将用户的请求平摊的分配到多个服务上，从而达到系统的HA。常见的负载均衡有软件Nginx，LVS，硬件 F5等。相应的在中间件，例如：dubbo和Spring Cloud中均给我们提供了负载均衡，SC的负载均衡算法可以自定义。

Ribbon负载均衡

Ribbon在工作时分成两步，第一步先选择 **Eureka Server**，它优先选择在**同一个Zone且负载较少的server**，第二步再根据**用户指定的策略**，在从**server**取到的**服务注册列表**中选择一个**地址**。其中Ribbon提供了三种策略：轮询、随机和根据响应时间加权。



Ribbon负载均衡

```
<dependency>  
    <groupId>org.springframework.cloud</groupId>  
    <artifactId>spring-cloud-starter-ribbon</artifactId>  
</dependency>
```

@Bean

@LoadBalanced

```
public RestTemplate restTemplate() {  
    return new RestTemplate();  
}
```

Spring Cloud的Feign

Feign是一种声明式、模板化的HTTP客户端。在Spring Cloud中使用Feign, 可以做到使用HTTP请求远程服务时能就像调用本地方法一样的体验, 开发者完全感知不到这是远程方法, 更感知不到这是个HTTP请求。

Feign使用示例

① Feign接口定义

```
1 package org.xujin.sc.feign.user.service;
2
3 import org.springframework.cloud.netflix.feign.FeignClient;
4 import org.springframework.web.bind.annotation.PathVariable;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestMethod;
7 import org.xujin.sc.feign.user.model.OrderModel;
8
9 @FeignClient(name = "sc-feign-first-provider")// [A]
10 public interface UserFeignService {
11
12     @RequestMapping(value = "/sc/order/{id}", method = RequestMethod.GET)// [B]
13     public OrderModel findOrderById(@PathVariable("id") Long id); // [C]
14
15 }
```

② 依赖注入使用

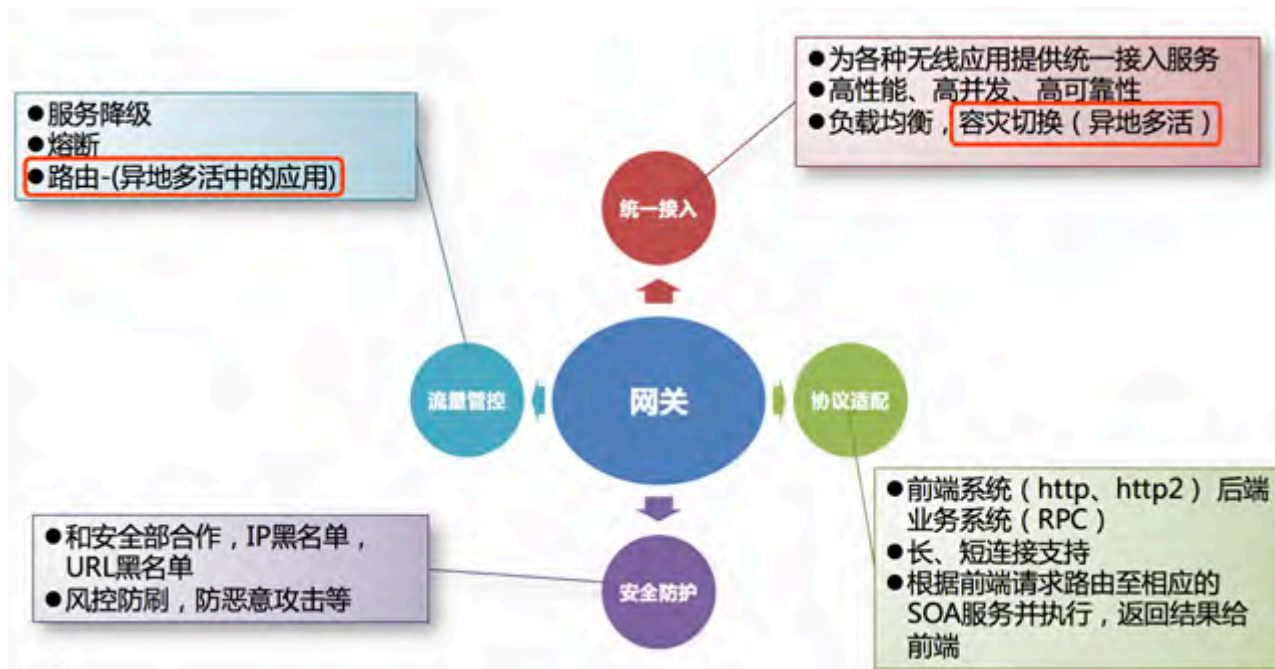
```
// 注入服务提供者, 远程的Http服务
@Autowired
private UserFeignService userFeignService;

// 服务消费者对位提供的服务
@GetMapping("/sc/user/{id}")
public OrderModel findByIdByEurekaServer(@PathVariable Long id) {
    return userFeignService.findOrderById(id);
}
```

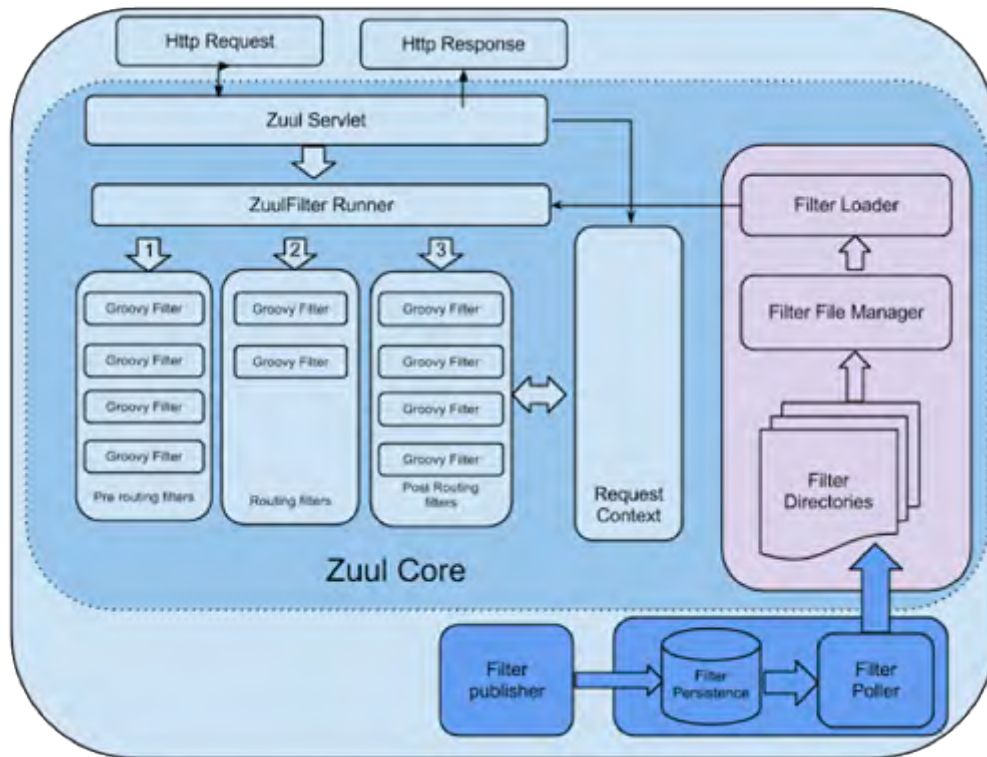
Spring Cloud Zuul

API Gateway (API GW / API 网关)，顾名思义，是出现在系统边界上的一个面向API的、串行集中式的**强管控服务**，这里的边界是**企业IT系统的边界**。Spring Cloud Zuul为Spring Cloud提供微服务网关。

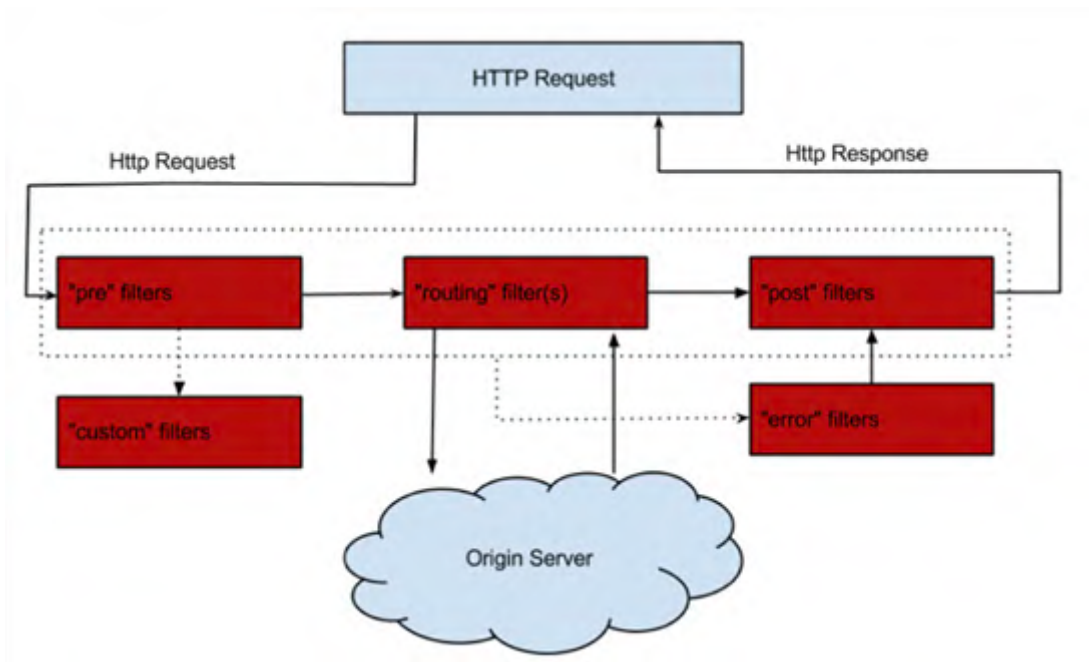
网关功能



Spring Cloud Zuul的架构图

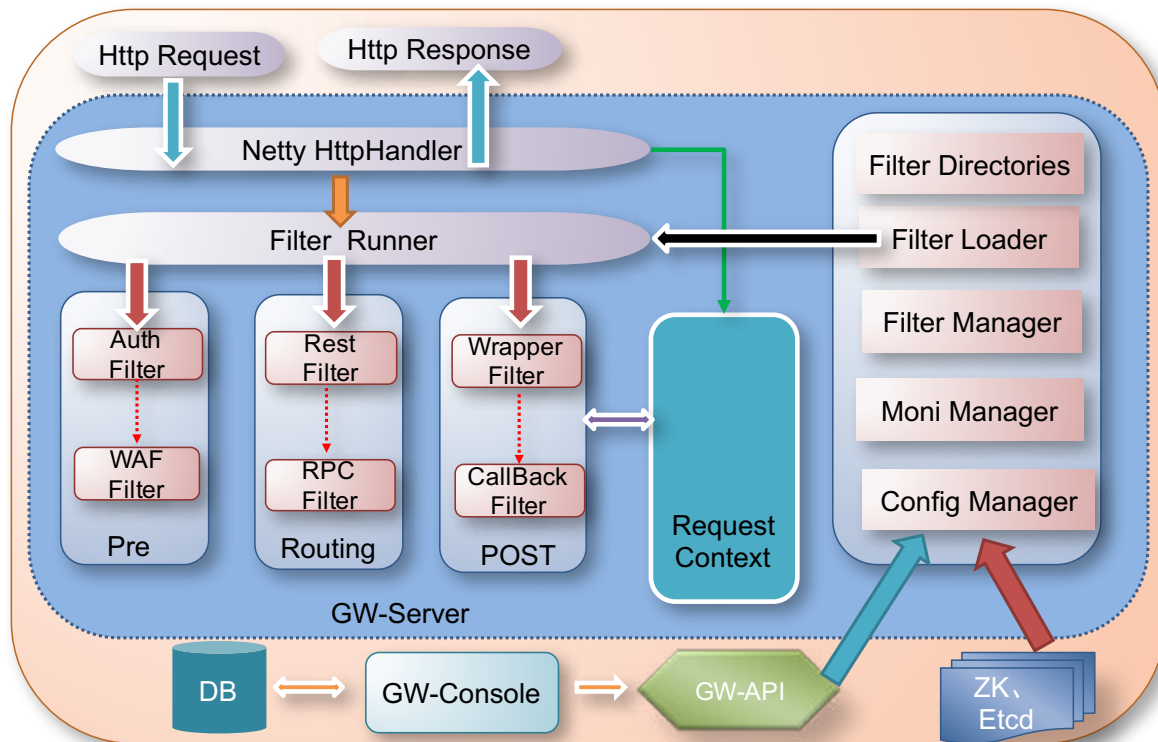


Spring Cloud Zuul的生命周期

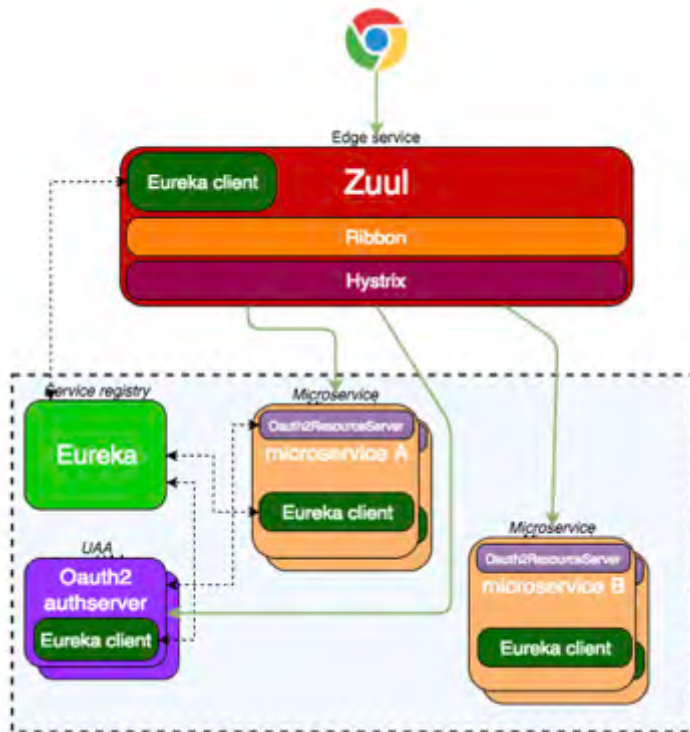


所有请求都必然按照pre-> route -> post的顺序执行。post返回response。

参考Zuul的架构设计



Zuul在Spring Cloud中处的位置



使用Zuul搭建网关


```
<dependencies>
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-zuul</artifactId>
  </dependency>
  <!-- 多了eureka starter -->
  <dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-eureka</artifactId>
  </dependency>
</dependencies>
```

<http://xujin.org/sc/sc-zuul-01/>

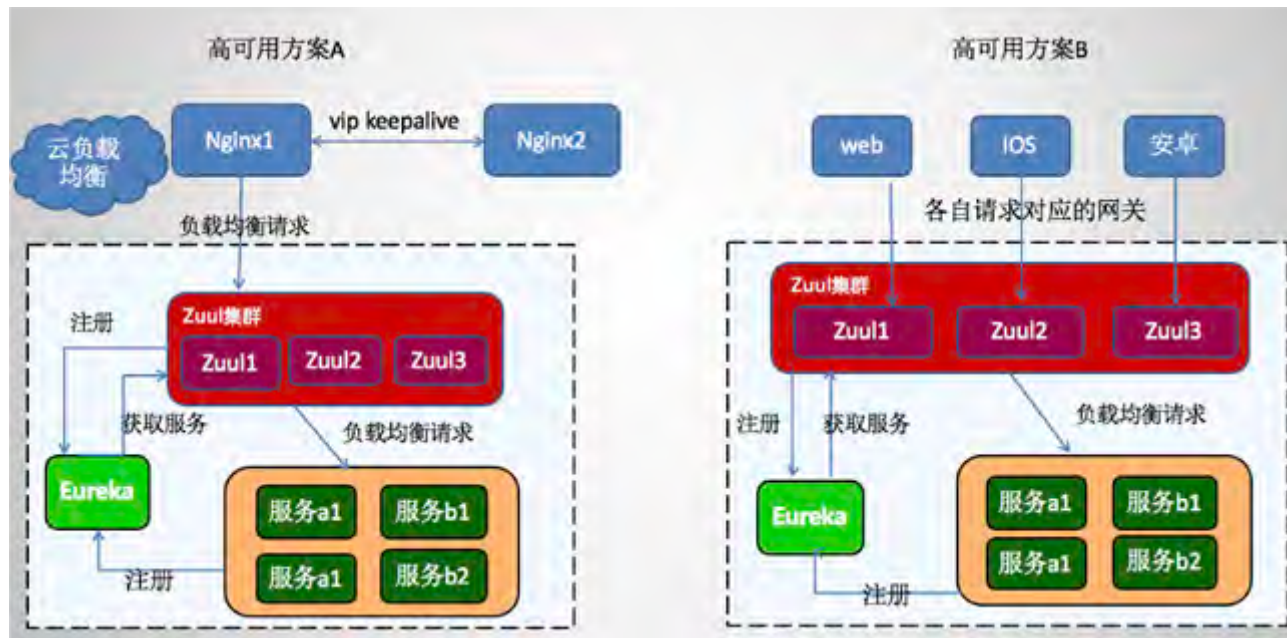
使用Zuul搭建网关

@EnableZuulProxy

```
1 server:
2   port: 8040
3 spring:
4   application:
5     name: sc-zuul-first-zuul
6 eureka:
7   client:
8     service-url:
9       defaultZone: http://localhost:8761/eureka/
10  instance:
11    prefer-ip-address: true
12
13 zuul:
14   routes:
15     sc-zuul-first-provider: /order/**
16
17
```



ZUUL的Ha方案



Spring-Cloud-Gateway



spring-cloud / spring-cloud-gateway

Watch 72 Star 130 Fork 36

Code Issues 21 Pull requests 0 Projects 0 Wiki Insights

A Gateway built on Spring Framework 5.0 and Spring Boot 2.0 providing routing and more. <http://cloud.spring.io>

java spring spring-boot spring-cloud microservices proxy api-gateway reactor

240 commits 6 branches 1 release 3 contributors Apache-2.0

Tag: v2.0.0.M1 New pull request Create new file Upload files Find file Clone or download

spencergibb	Update SNAPSHOT to 2.0.0.M1	Latest commit 389cab1 12 days ago
.mvn	Updates for reactor-bom Bismuth	3 months ago
docs	Update SNAPSHOT to 2.0.0.M1	12 days ago
spring-cloud-gateway-core	Update SNAPSHOT to 2.0.0.M1	12 days ago
spring-cloud-gateway-dependencies	Update SNAPSHOT to 2.0.0.M1	12 days ago
spring-cloud-gateway-mvc	Update SNAPSHOT to 2.0.0.M1	12 days ago
spring-cloud-gateway-sample	Update SNAPSHOT to 2.0.0.M1	12 days ago
spring-cloud-starter-gateway	Update SNAPSHOT to 2.0.0.M1	12 days ago

- Java 8/Spring 5/Boot 2
- WebFlux/Reactor
- HTTP/2 and Websockets
- Finchley Release Train (Q4 2017)

Spring Cloud的配置中心

Spring Cloud Config就是我们通常意义上的配置中心,
把应用原本放在本地文件的配置抽取出来放在中心服务器,
器, 从而能够提供更好的管理、发布能力。

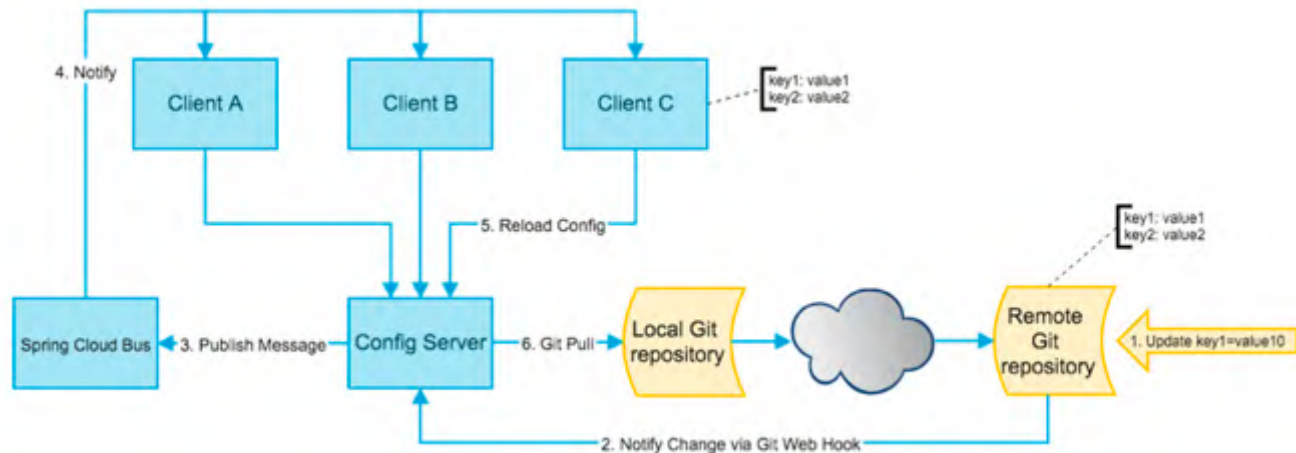
配置中心的功能



Spring Cloud Config作为配置中心不是很适合，因此技术选型会选择携程配置中心Apollo

<https://github.com/ctripcorp/apollo>

Spring Cloud Config工作流程



快速使用Spring Cloud Config

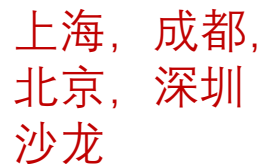
```
package org.xujin.sc.config.server;  
  
import org.springframework.boot.SpringApplication;  
  
@EnableConfigServer  
@SpringBootApplication  
public class SpringCloudConfigServer {  
    public static void main(String[] args) {  
        SpringApplication.run(SpringCloudConfigServer.class, args);  
    }  
}
```

添加依赖，使用@EnableConfigServer注解

快速使用Spring Cloud Config

```
server:
  port: 8040
spring:
  application:
    name: sc-config-server
  cloud:
    config:
      server:
        git:
          uri: https://github.com/SoftwareKing/spring-cloud-study/
          search-paths: sc-config-conf
          username: # !
          password: # !
```

在application.yml配置如上信息



Spring Cloud中国社区文档:<http://docs.springcloud.cn>

目录

1 Spring Cloud概述

2 Spring Cloud国内使用情况

3 使用Spring Cloud搭建微服务应用

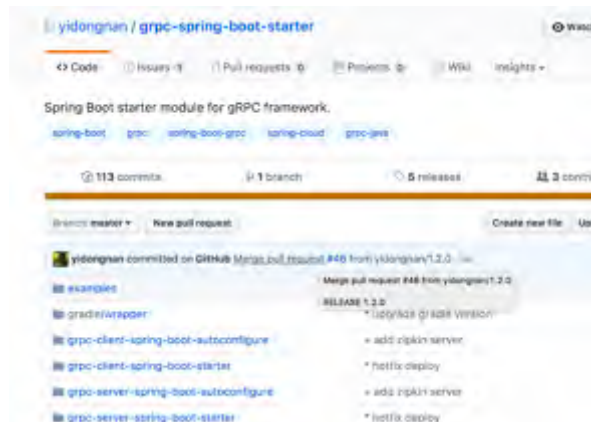
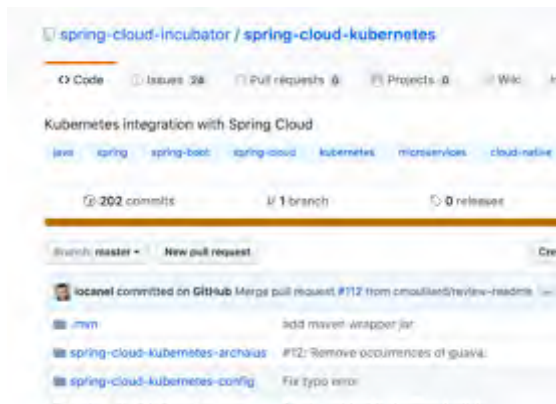
➔ 4 展望Spring Cloud未来

5 Q&A

Spring Cloud-百花齐放

SPRING CLOUD GATEWAY

- Java 8/Spring 5/Boot 2
- WebFlux/Reactor
- HTTP/2 and Websockets
- **Finchley Release Train (Q4 2017)**



目录

1

Spring Cloud概述

2

Spring Cloud国内使用情况

3

使用Spring Cloud搭建微服务应用

4

展望Spring Cloud未来



5

Q&A

Q&A

欢迎加我
微信互撩





高效运维社区
GreatOPS Community



会议

培训

咨询

- 8月18日 DevOpsDays 上海
- 全年 DevOps China 巡回沙龙
- 11月17日 DevOps金融上海

- EXIN DevOps Master 认证培训
- DevOps 企业内训
- DevOps 公开课
- 互联网运维培训
- 企业DevOps 实践咨询
- 企业运维咨询



商务经理：刘静女士
电话 / 微信：13021082989
邮箱：liujing@greatops.com



Thanks

荣誉出品

高效运维社区

国际最佳实践管理联盟



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

