课时6

API 网关服务-Zuul

1. 网关的必要性
2. Zuul简介
3. Zuul自定义过滤器
4. Zuul容错与回退
5. Zuul使用小经验
6. Zuul控制路由实例选择

网关是什么？

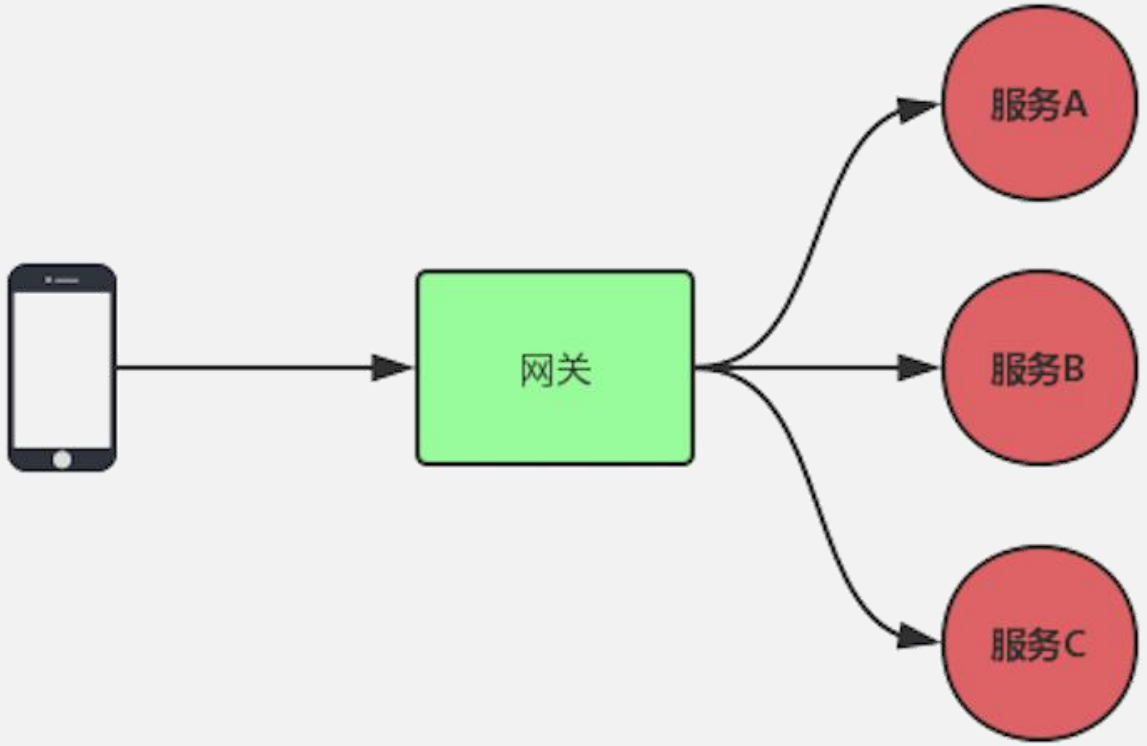
拉勾 LAGOU.COM

API网关

是对外服务的一个入口，隐藏了内部架构的实现

是微服务架构中必不可少的一个组件

可以为我们管理大量的API接口，对接客户，协议适配，安全认证，路由转发，流量限制，日志监控，防止爬虫，灰度发布等等功能



网关的必要性

拉勾
LAGOU.COM

动态路由

灰度发布

请求监控

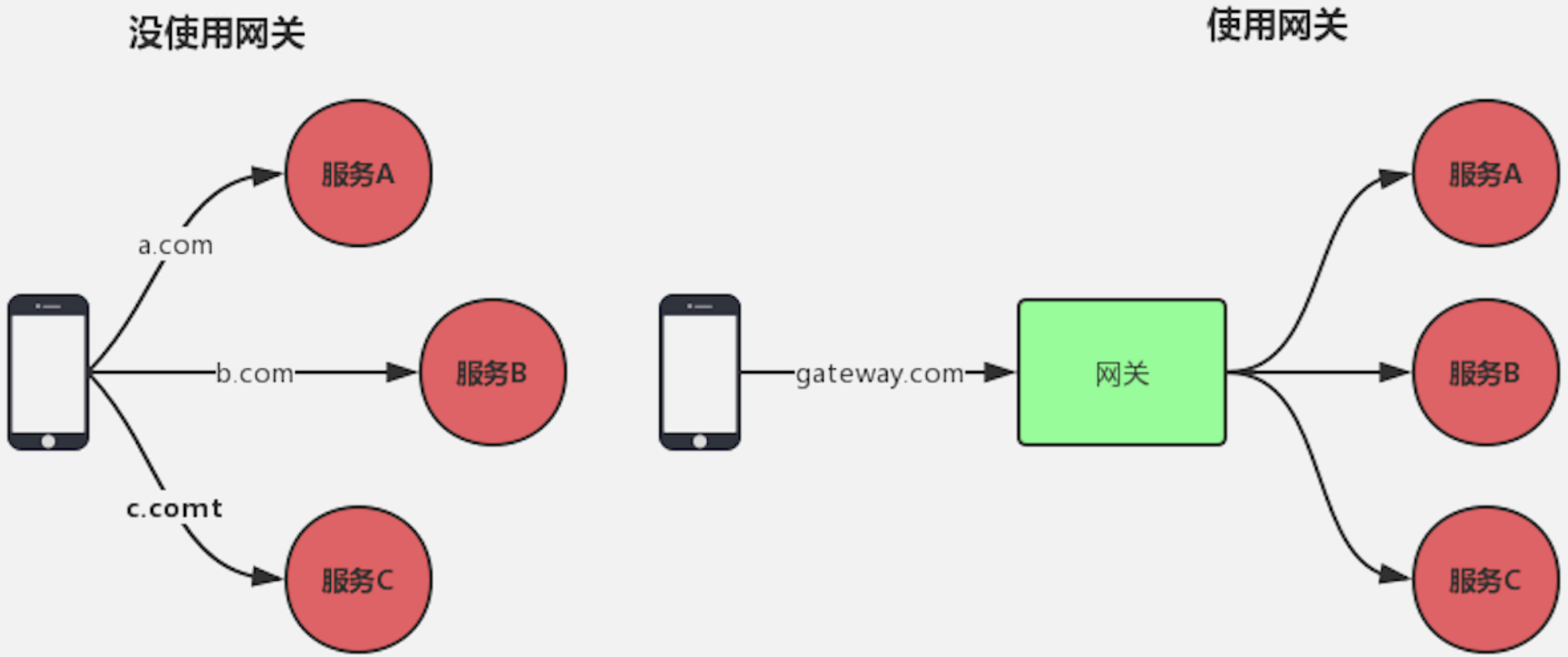
压力测试

认证鉴权

网关的必要性-动态路由

拉勾 LAGOU.COM

- 动态的将客户端的请求路由到后端不同的服务

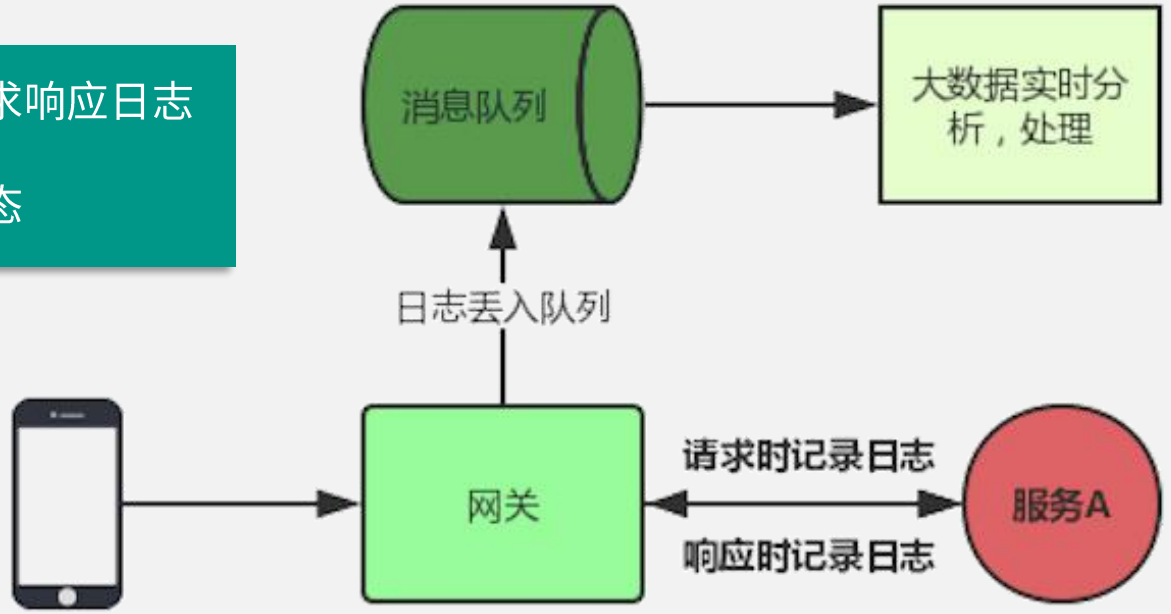


网关的必要性-请求监控

拉勾 LAGOU.COM

可以对整个系统的请求进行监控，记录详细的请求响应日志

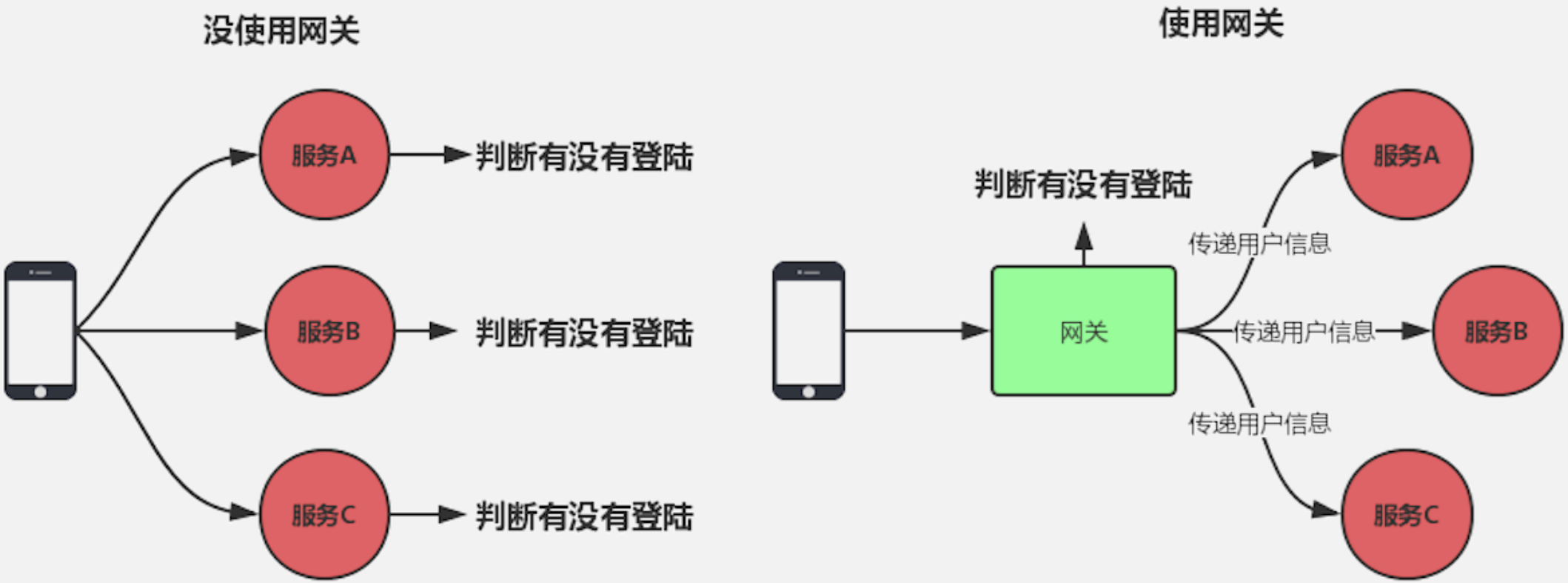
可以实时的统计出当前系统的访问量以及监控状态



网关的必要性-认证鉴权

拉勾 LAGOU.COM

- 对每一个访问的请求做认证，拒绝非法的请求，保护好后端的服务

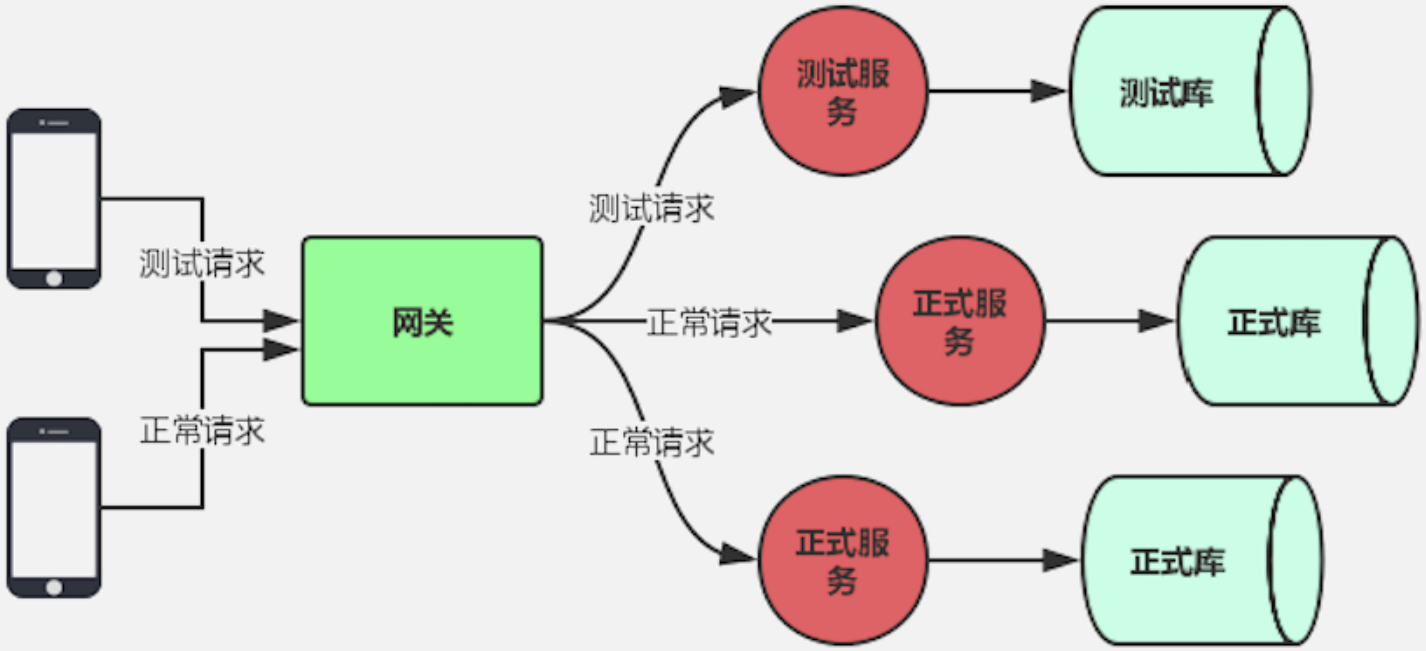


网关的必要性-压力测试

拉勾 LAGOU.COM

压力测试

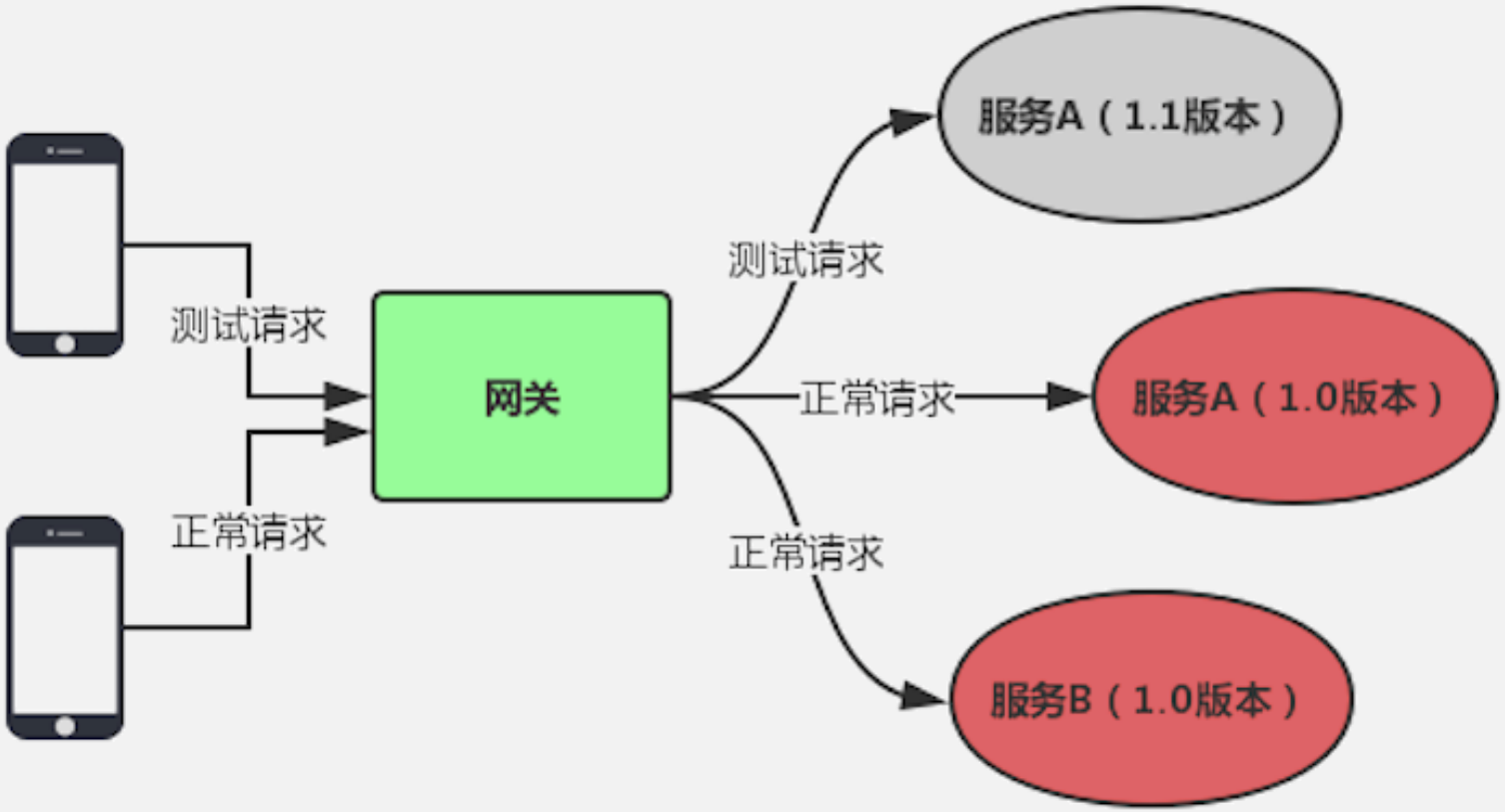
是一项很重要的工作，像一些电商公司需要模拟更多真实的用户并发量来保证大促时系统的稳定，通过Zuul可以动态的转发到后端服务的集群中，还可以识别测试流量和真实流量，用来做一些特殊处理



网关的必要性-灰度发布

拉勾 LAGOU.COM

- 灰度发布可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以保证其影响度



Zuul简介

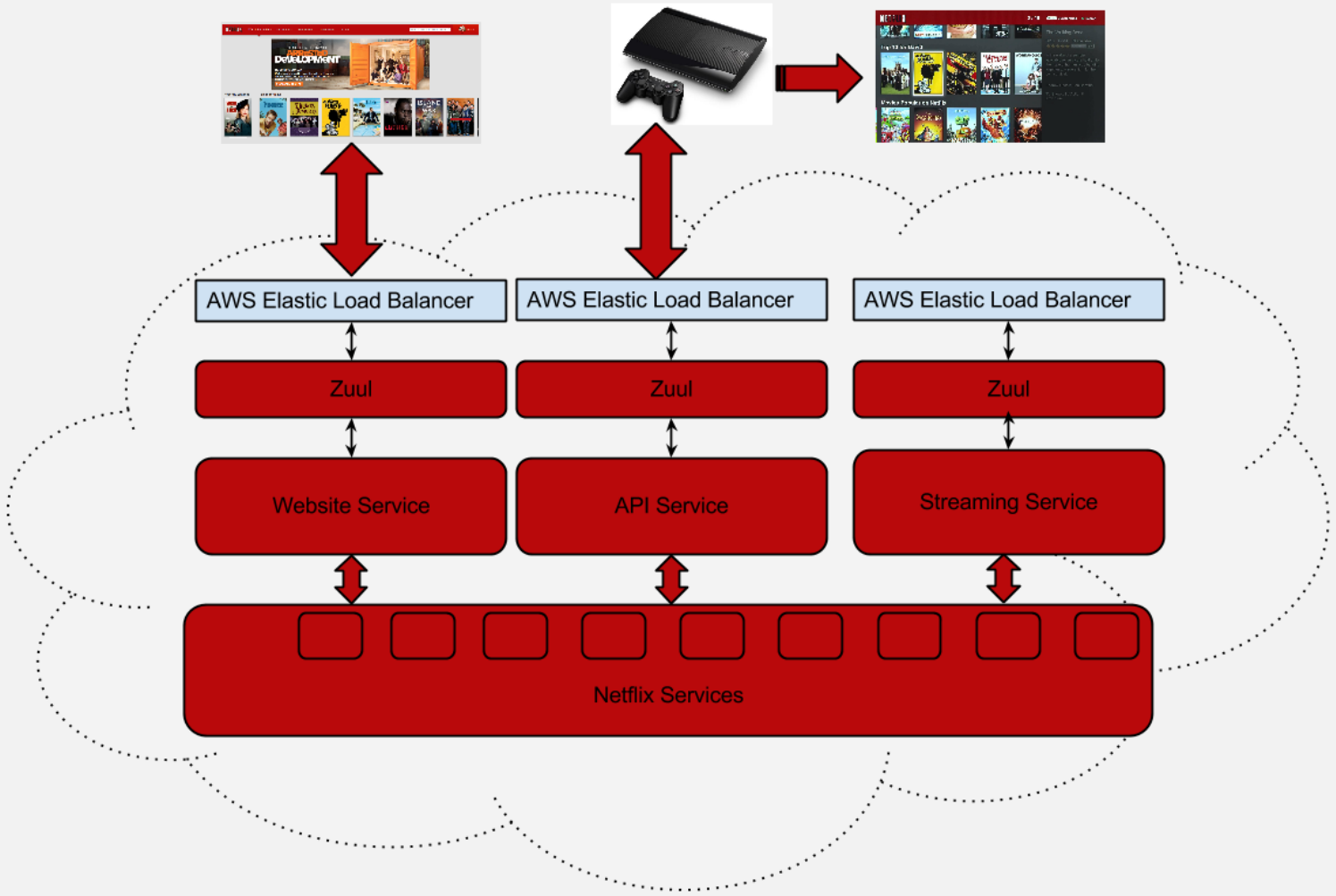
拉勾 LAGOU.COM

Zuul

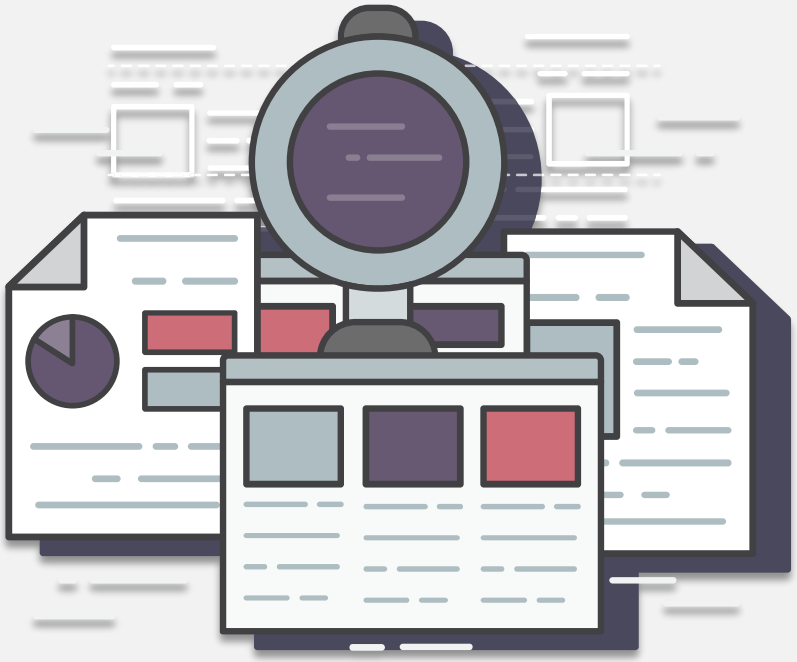
是Netflix OSS中的一员，一个基于JVM路由和服务端的负载均衡器

提供路由，监控，弹性，安全等服务的框架

Zuul能够与Eureka,Ribbon,Hystrix等组件配合使用



Zuul过滤器



pre

1. 可以在请求被路由之前调用
2. 适用于身份认证的场景，认证通过后再继续执行下面的流程

route

1. 在路由请求时候被调用
2. 适用于灰度发布这种场景下，在将要路由的时候可以做一些自定义的逻辑

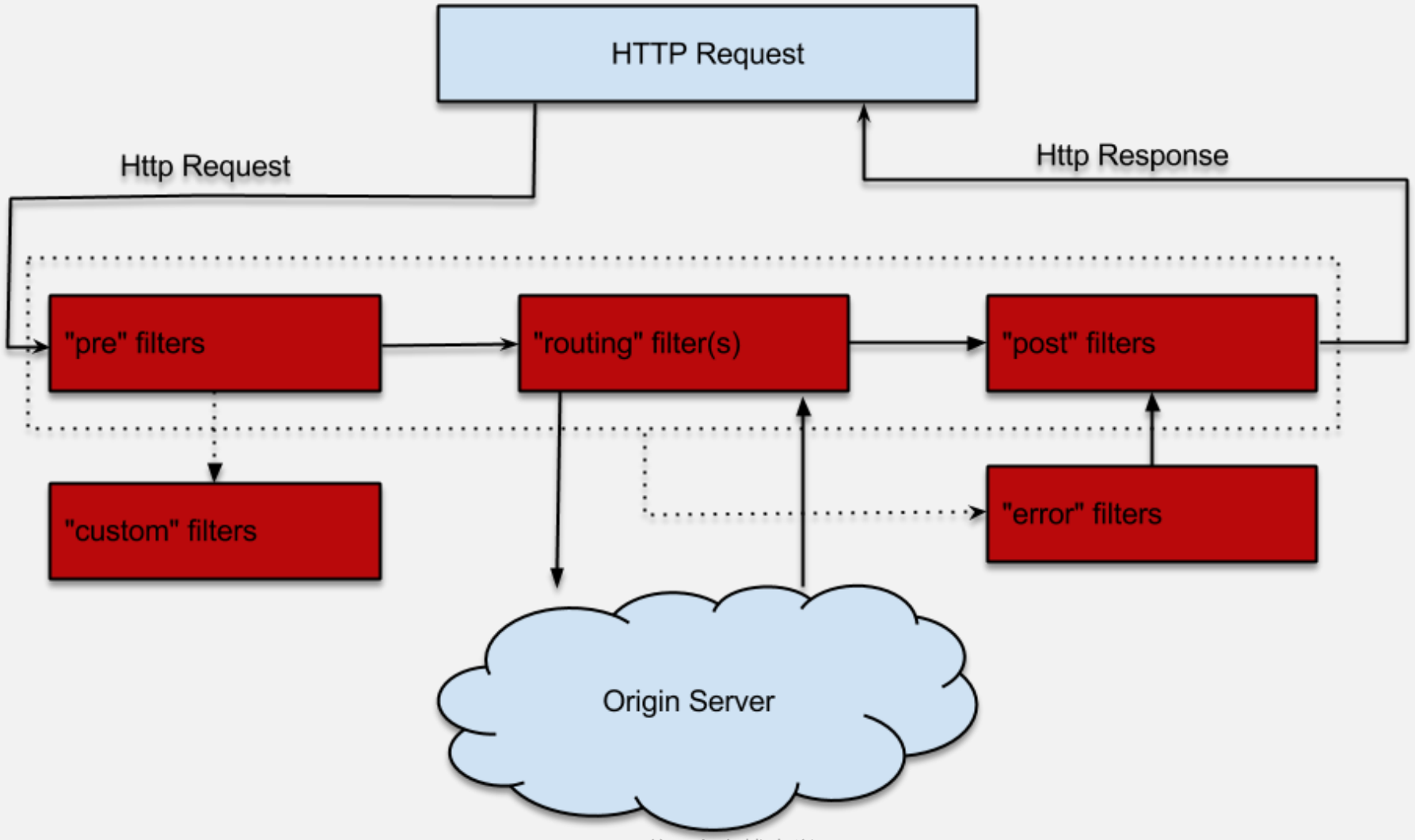
post

1. 在route和error过滤器之后被调用
2. 这种过滤器将请求路由到达具体的服务之后执行
3. 适用于添加响应头，记录响应日志等应用场景

error

1. 处理请求时发生错误时被调用
2. 执行过程中发送错误时会进入error过滤器，可以用来统一记录错误信息

Zuul请求生命周期



Zuul路由

拉勾
LAGOU.COM

URL路由,不具备负载均衡

配置多个URL负载均衡

为所有请求加前缀

基于Eureka代理服务

Zuul自定义过滤器

拉勾 LAGOU.COM

```java
@Component
public class MyFilter extends ZuulFilter {

    @Override
    public boolean shouldFilter() {
        return true;
    }

    @Override
    public Object run() throws ZuulException {
        return null;
    }

    @Override
    public String filterType() {
        return "pre";
    }

    @Override
    public int filterOrder() {
        return 0;
    }

}
```

- shouldFilter
是否执行该过滤器，true为执行，false为不执行，
这个也可以利用配置中心来做，达到动态的开启和关闭过滤器

- filterType
过滤器类型，可选值有pre、route、post、error

- filterOrder
过滤器的执行顺序，数字越小，优先级越高

- run
业务逻辑

Zuul容错与回退

拉勾 LAGOU.COM

- Spring Cloud中，Zuul默认整合了Hystrix，当后端服务异常时可以为Zuul添加回退功能，返回默认的数据给客户端

- 隔离方式配置
  zuul.ribbon-isolation-strategy=THREAD|SEMAPHORE

Zuul小经验分享-内置端点

拉勾 LAGOU.COM

/actuator/filters


```
{
  "error": [
    {
      "class": "org.springframework.cloud.netflix.zuul.filters.post.SendErrorFilter",
      "order": 0,
      "disabled": false,
      "static": true
    }
  ]
}
```

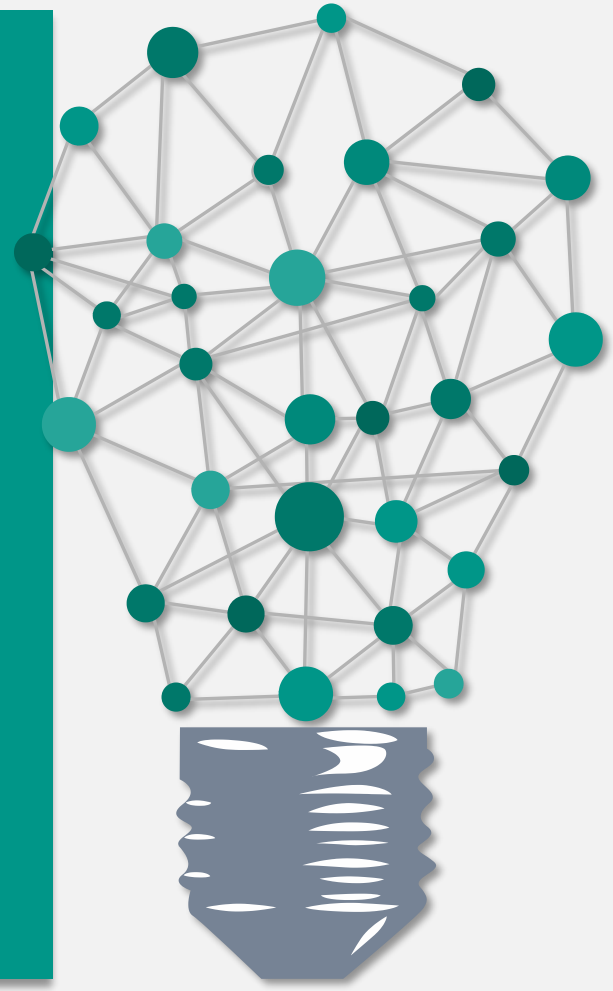
Zuul小经验分享-文件上传

拉勾 LAGOU.COM

1. 上传限制修改
spring.servlet.multipart.max-file-size=1000Mb
spring.servlet.multipart.max-request-size=1000Mb

2.绕过Spring DispatcherServlet进行上传大文件
正常的地址
http://localhost:2103/zuul-file/file/upload
绕过的地址
http://localhost:2103/zuul/zuul-file/file/upload

3.超时时间
ribbon.ConnectTimeout=3000
ribbon.ReadTimeout=60000
# Zuul的Hystrix隔离模式为线程才配置
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds=60000



Zuul小经验分享-请求响应输出

拉勾
LAGOU.COM

文章参考：

http://cxytiandi.com/blog/detail/20343

http://cxytiandi.com/blog/detail/20529

Zuul小经验分享-Zuul Debug

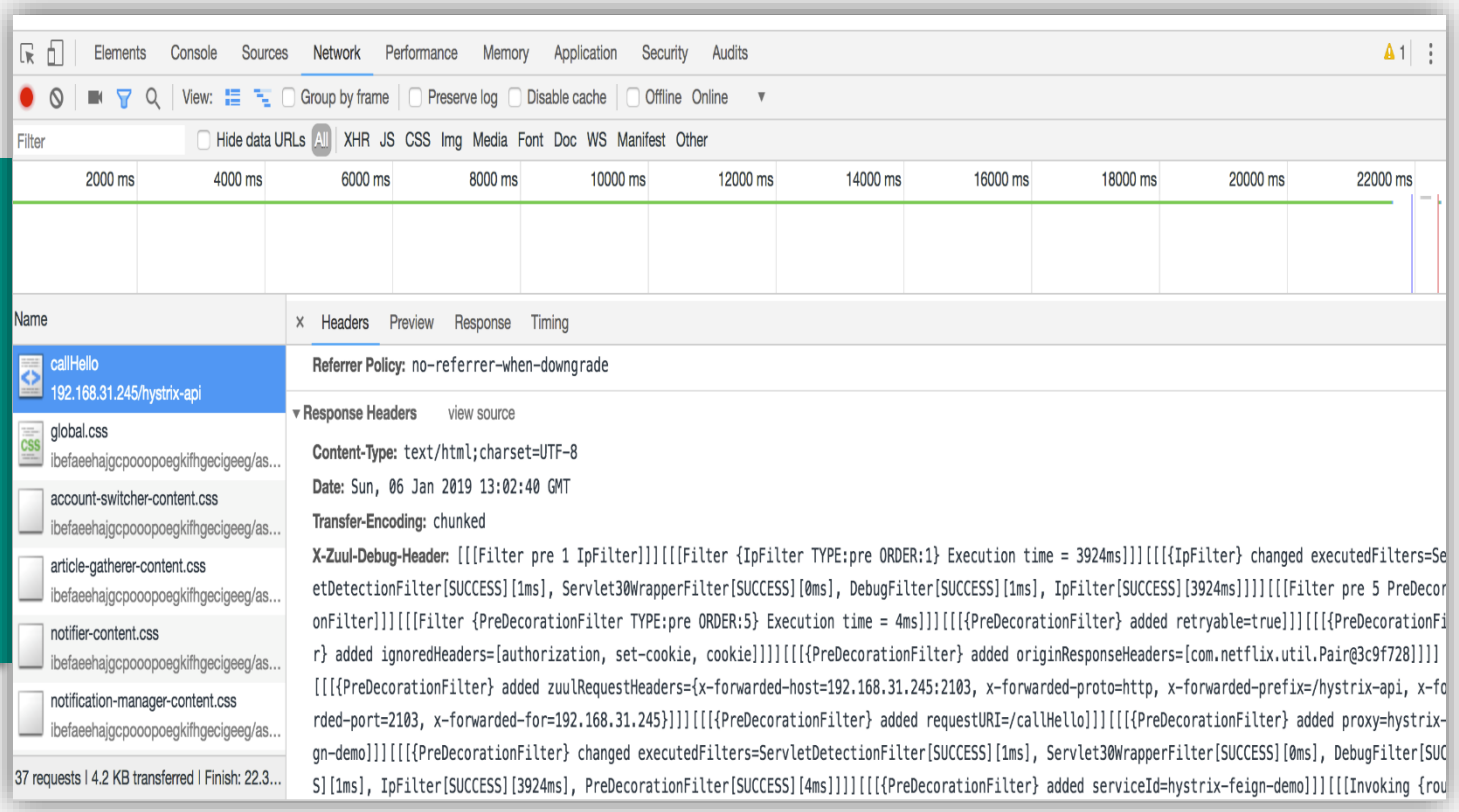
拉勾 LAGOU.COM

增加配置：

zuul.include-debug-header=true

请求地址增加参数debug=true：

http://xxx.com/api/user?debug=true



Zuul小经验分享-跨域配置

拉勾
LAGOU.COM

```java
@Configuration
public class GateWayCorsConfig {

    @Bean
    public CorsFilter corsFilter() {
        final UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        final CorsConfiguration corsConfiguration = new CorsConfiguration();
        corsConfiguration.addAllowedHeader("*");
        corsConfiguration.addAllowedOrigin("*");
        corsConfiguration.addAllowedMethod("*");
        source.registerCorsConfiguration("/**", corsConfiguration);
        return new CorsFilter(source);
    }

}
```

Zuul小经验分享-关闭zuul全局路由转发

拉勾
LAGOU.COM

禁用默认为所有服务代理转发

zuul.ignored-services=*

忽略包含student的路径

zuul.ignoredPatterns=/**/student/**

Zuul小经验分享-动态过滤器

拉勾 LAGOU.COM

Zuul 支持过滤器动态修改加载功能，Filter需要使用Groovy编写才可以被动态加载

```
// 定时从一个目录加载Groovy的Filter
FilterLoader.getInstance().setCompiler(new GroovyCompiler());
try {
    FilterFileManager.setFilenameFilter(new GroovyFileFilter());
    FilterFileManager.init(20, "/Users/yinjihuan/Downloads/filters");
} catch (Exception e) {
    throw new RuntimeException();
}
```

Zuul扩展之控制路由实例选择

1.定义Ribbon负载均衡策略

2.实现实例选择逻辑

3.全局应用

Next：课时7《分布式配置中心-Apollo》