为什么要网关?

微服务下一个系统被拆分为多个服务,但是像 安全认证,流量控制,日志,监控等功能是每个服务都需要的,没有网关的话,我们就需要在每个服务中单独实现,这使得我们做了很多重复的事情并且没有一个全局的视图来统一管理这些功能。

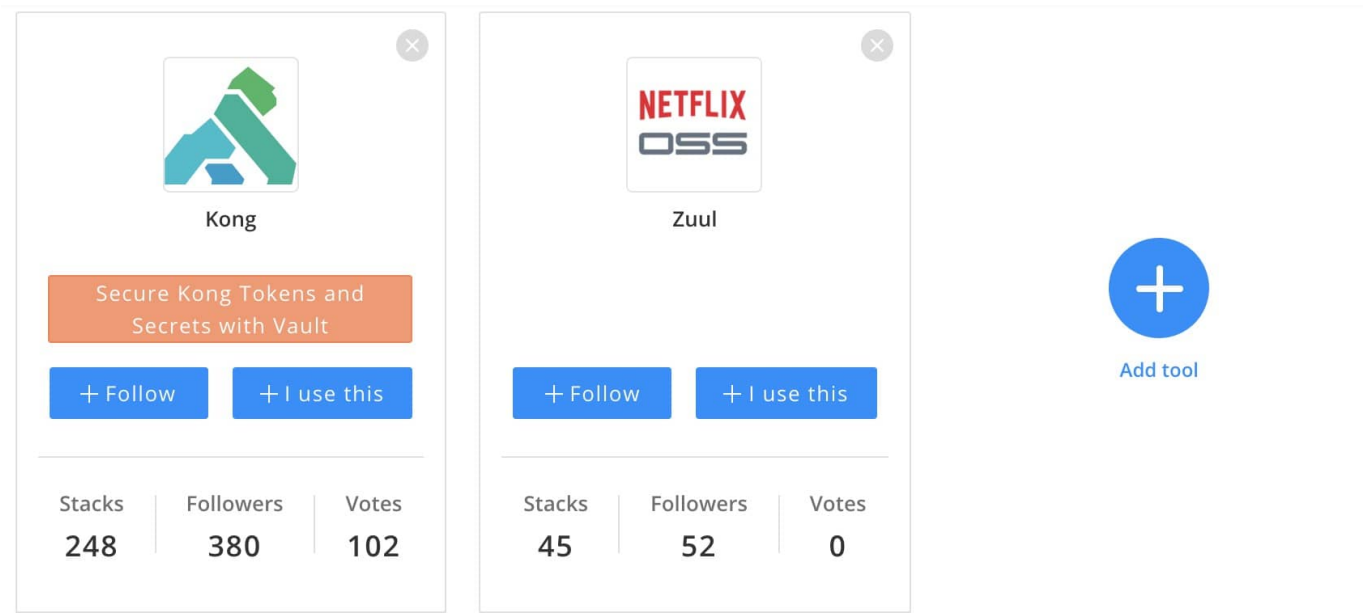综上:**一般情况下,网关一般都会提供请求转发、安全认证(身份/权限认证)、流量控制、负载均衡、容灾、日志、监控这些功能。**

上面介绍了这么多功能实际上网关主要做了一件事情:**请求过滤** 。权限校验、流量控制这些都可以通过过滤器实现,请求转也是通过过滤器实现的。

你知道有哪些常见的网关系统?

我所了解的目前经常用到的开源 API 网关系统有:

1. Kong
2. Netflix zuul

下图来源:https://www.stackshare.io/stackups/kong-vs-zuul 。

Kong vs Zuul: What are the differences?

What is Kong? *Open Source Microservice & API Management Layer*. Kong is a scalable, open source API Layer (also known as an API Gateway, or API Middleware). Kong controls layer 4 and 7 traffic and is extended through Plugins, which provide extra functionality and services beyond the core platform.

What is Zuul? *An edge service that provides dynamic routing, monitoring, resiliency, security, and more*. It is the front door for all requests from devices and websites to the backend of the Netflix streaming application. As an edge service application, It is built to enable dynamic routing, monitoring, resiliency, and security. Routing is an integral part of a microservice architecture.

Kong and Zuul can be categorized as **"Microservices"** tools.

Kong is an open source tool with **22.9K** GitHub stars and **2.83K** GitHub forks. Here's a link to Kong's open source repository on GitHub.

Checkr, **Policygenius**, and **Decision6** are some of the popular companies that use Kong, whereas Zuul is used by **PeopleKeep**, **Monkey Exchange**, and **Wealth Systems**. Kong has a broader approval, being mentioned in **73** company stacks & **112** developers stacks; compared to Zuul, which is listed in **10** company stacks and **8** developer stacks.

可以看出不论是社区活跃度还是 Star数， Kong 都是略胜一筹。总的来说，Kong 相比于 Zuul 更加强大并且简单易用。Kong 基于 Openresty ，Zuul 基于 Java。

> OpenResty（也称为 ngx_openresty）是一个全功能的 Web 应用服务器。它打包了标准的 Nginx 核心，很多的常用的第三方模块，以及它们的大多数依赖项。
>
> 通过揉和众多设计良好的 Nginx 模块，OpenResty 有效地把 Nginx 服务器转变为一个强大的 Web 应用服务器，基于它开发人员可以使用 Lua 编程语言对 Nginx 核心以及现有的各种 Nginx C 模块进行脚本编程，构建出可以处理一万以上并发请求的极端高性能的 Web 应用。——OpenResty

另外， Kong 还提供了插件机制来扩展其功能。

比如、在服务上启用 Zipkin 插件

```
$ curl -X POST http://kong:8001/services/{service}/plugins \
    --data "name=zipkin"  \
    --data "config.http_endpoint=http://your.zipkin.collector:9411/api/v2/spans" \
    --data "config.sample_ratio=0.001"
```

ps:这里没有太深入去探讨，需要深入了解的话可以自行查阅相关资料。