什么是API 网关？为什么需要 API网关？

大家好，我是猿java。

今天我们来聊一聊，什么是 API 网关？它有什么作用？为什么我们需要它？

什么是API

在维基百科中，网关的定义是这样的：

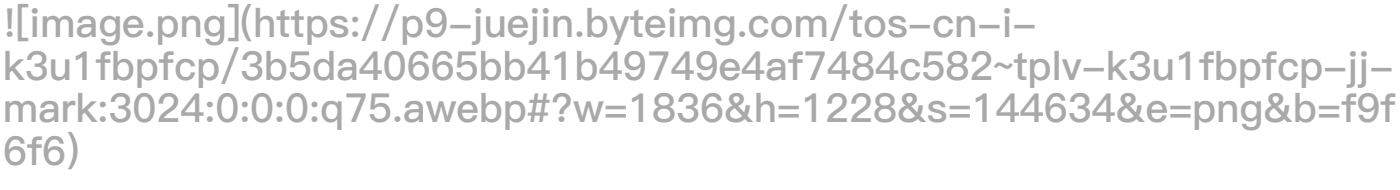
> 在计算机网络中，网关（Gateway）是转发其他服务器通信数据的服务器，接收从客户端发送来的请求时，它就像自己拥有资源的源服务器一样对请求进行处理。有时客户端可能都不会察觉，自己的通信目标是一个网关。

从定义可以看出，网关也是一组服务器，它位于客户端和服务器之间，是客户端请求进入服务器的唯一入口，

API的作用

如下图，API 网关提供 几个重要的功能：

1. 身份验证和安全策略实施；
2. 负载平衡和短路；
3. 协议转换和服务发现；
4. 监控、日志记录、分析和计费；
5. 缓存；



典型流程分析

下面我们通过客户端向服务器发起一个HTTP请求这个经典的流程来讲解API网关及其重要的功能。

第一步：客户端向 API 网关发送请求

客户端向服务器发起一个请求，该请求通常是基于 HTTP协议，它可以是 REST、GraphQL 或其他一些更高级别的抽象。如下图：

第二步： API 网关验证 HTTP 请求

API网关收到客户端的请求后，会对 HTTP请求中的参数等进行校验，如下图：

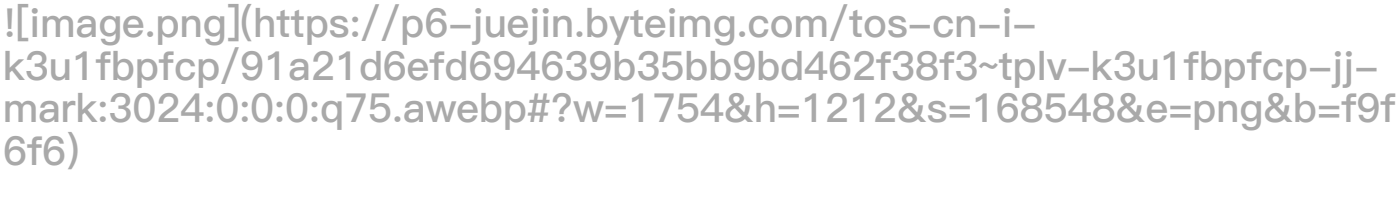
第三步：IP 黑白名单校验

为了安全，可以在API网关设置IP黑白单，标志允许和不允许访问服务器的IP，API 网关根据 IP黑白列表来允许和拒绝调用者的IP地址。如下图：

同时，API网关还可以针对IP 地址和 HTTP标头等属性执行基本的速率限制检查。例如，它可以拒绝来自超过一定速率的 IP地址的请求。如下图：

第四步：身份验证和授权

API 网关将 HTTP请求传递给身份提供商以进行身份验证和授权。API 网关从提供商处接收经过身份验证的会话，其中包含允许请求执行的操作范围。如下图：



认证是验证用户或客户端身份的过程，用于确认一个实体是否为其所声称的那个实体。在API的上下文中，这意味着确保请求方是合法的用户或客户端，并且有权访问所请求的资源或服务。

常见的认证方式包括：

* 基本认证（Basic Authentication）：客户端在请求头中使用Base64编码的用户名和密码进行认证。虽然简单易用，但不是最安全的认证方法，因为凭据会以明文形式在请求中传输，容易被拦截和解码。
* 令牌认证（Token Authentication）：客户端在请求头中使用特定的令牌（Token）进行认证。令牌通常由身份验证后的服务颁发给客户端，有效期有限，并且在每次请求中传递。相对于基本认证，令牌认证更安全，因为令牌通常不包含敏感信息，且可以通过HTTPS加密进行传输。
* OAuth认证：OAuth是一种用于授权的开放标准。它允许用户授权第三方应用访问他们存储在另一个服务提供者上的资源，而无需提供他们的登录凭据。OAuth通常用于允许用户通过第三方身份验证进行访问。

授权是在认证成功后，决定用户或客户端是否有权访问特定资源或执行特定操

作的过程。它定义了用户在系统中的权限和角色，并根据这些权限来限制对资源的访问。

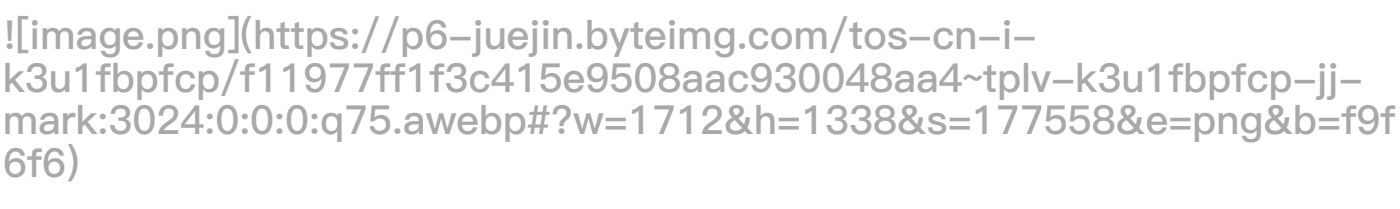
常见的授权方式包括：

* 角色-Based授权（Role-Based Authorization）：在角色-Based授权中，用户被分配到不同的角色，每个角色有不同的权限。例如，管理员角色可能有权访问所有资源，而普通用户角色可能只有限制的权限。
* 资源-Based授权（Resource-Based Authorization）：在资源-Based授权中，访问权限是直接授予特定资源的，而不是基于角色。每个资源都可以定义其自己的权限规则，决定哪些用户或角色可以访问它。
* 访问令牌（Access Token）：在OAuth认证中，访问令牌是用于授权的重要组成部分。访问令牌包含有关用户或客户端的授权信息，以及所被授权访问的资源和权限。

综合来说，认证用于确认用户或客户端的身份，而授权用于确定用户或客户端是否有权访问特定资源或执行特定操作。这两个步骤共同确保只有合法且有权访问的用户或客户端可以使用API，并保护系统免受未经授权的访问。在API网关中，认证和授权是非常重要的功能，因为它们直接影响到整个系统的安全性和数据的保护。

第五步：流量控制和限流

客户端请求的身份验证通过后，API网关可以做更高级别的流量控制和限流。如下图：



流量控制和限流是在API网关中用于管理和控制请求流量的重要概念。它们有助于维护后端服务的稳定性，防止过载，并提供更好的性能和可靠性。下面详细解释这两个概念：

流量控制（Rate Limiting）是指对请求的速率进行控制，以限制客户端对API的请求频率。这个过程可以确保后端服务不会受到过多请求的压力，避免服务器资源过度消耗，导致系统崩溃或响应缓慢。 常见的流量控制方法包括：

* 固定窗口计数器（Fixed Window Counter）：在固定时间窗口内（例如每分钟），对每个客户端或API密钥的请求计数。超过预设的请求数量限制时，拒绝额外的请求或延迟响应。
* 滑动窗口计数器（Sliding Window Counter）：类似于固定窗口计数器，但窗口是滑动的，允许更灵活地控制请求速率。
* 令牌桶算法（Token Bucket Algorithm）：通过将令牌存放在桶中来控制请求速率。每个令牌代表一个请求，桶有一个固定容量。每当有请求时，一个令牌将被消耗，当桶中没有令牌时，则限制进一步的请求。

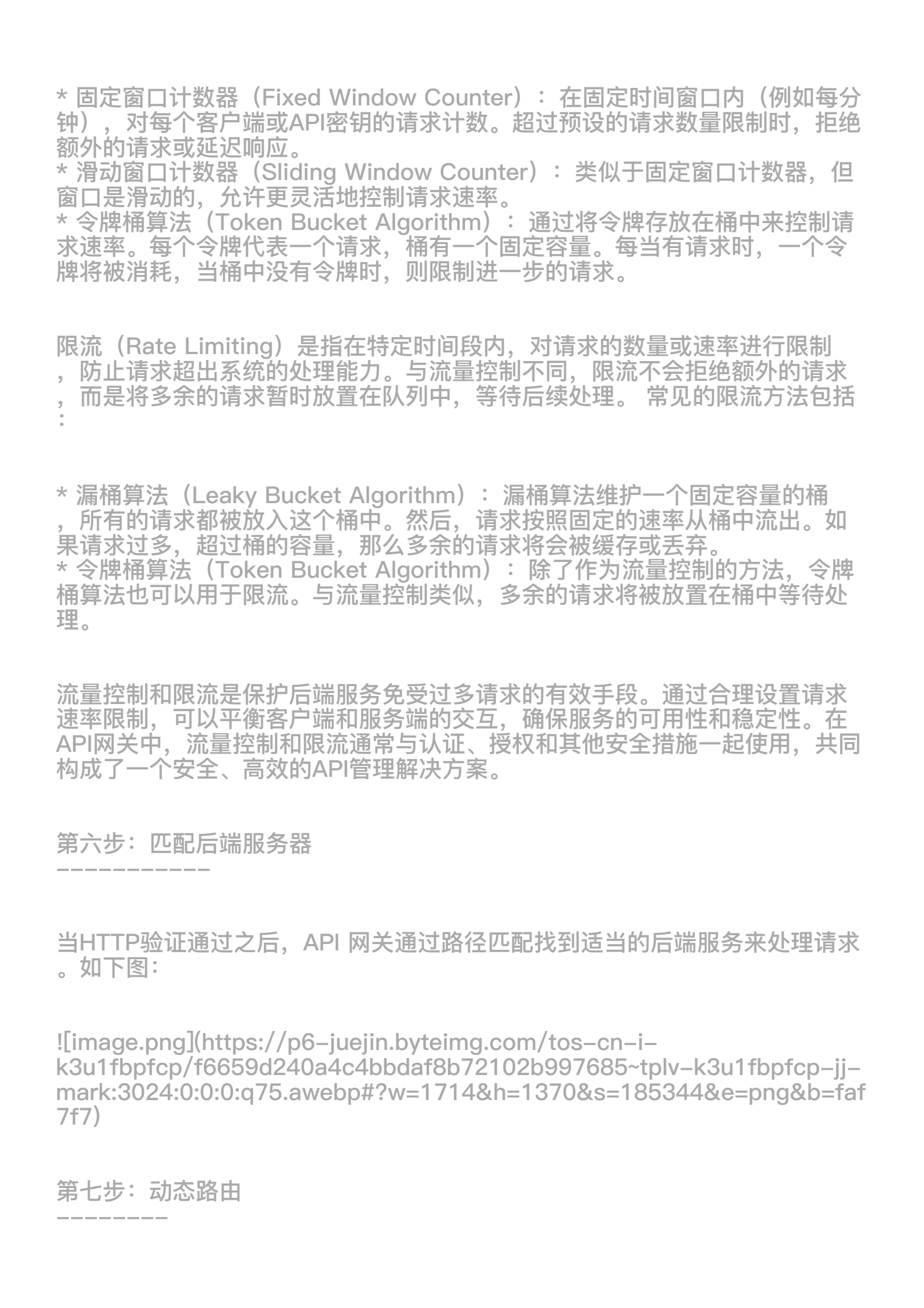
限流（Rate Limiting）是指在特定时间段内，对请求的数量或速率进行限制，防止请求超出系统的处理能力。与流量控制不同，限流不会拒绝额外的请求，而是将多余的请求暂时放置在队列中，等待后续处理。 常见的限流方法包括：

* 漏桶算法（Leaky Bucket Algorithm）：漏桶算法维护一个固定容量的桶，所有的请求都被放入这个桶中。然后，请求按照固定的速率从桶中流出。如果请求过多，超过桶的容量，那么多余的请求将会被缓存或丢弃。
* 令牌桶算法（Token Bucket Algorithm）：除了作为流量控制的方法，令牌桶算法也可以用于限流。与流量控制类似，多余的请求将被放置在桶中等待处理。

流量控制和限流是保护后端服务免受过多请求的有效手段。通过合理设置请求速率限制，可以平衡客户端和服务端的交互，确保服务的可用性和稳定性。在API网关中，流量控制和限流通常与认证、授权和其他安全措施一起使用，共同构成了一个安全、高效的API管理解决方案。

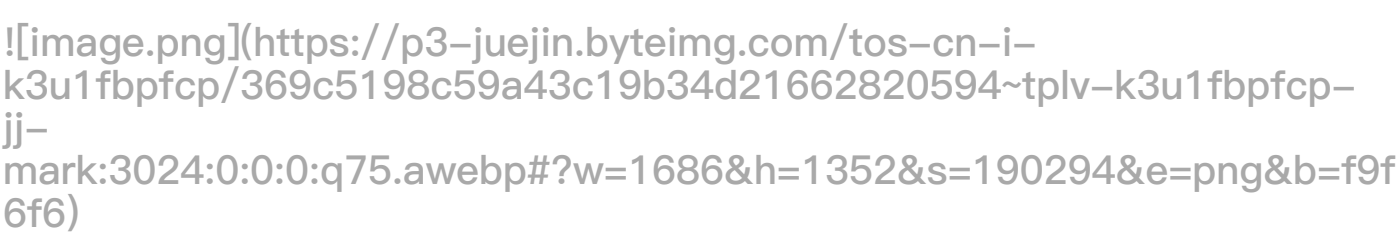
第六步：匹配后端服务器

当HTTP验证通过之后，API 网关通过路径匹配找到适当的后端服务来处理请求。如下图：



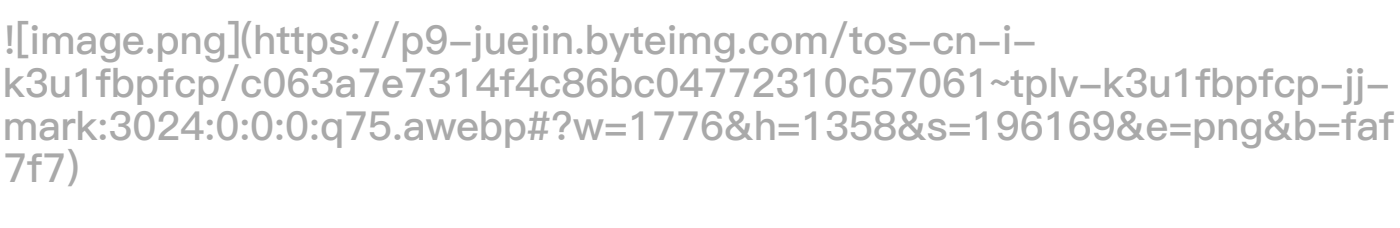
第七步：动态路由

匹配到对应到服务器之后，则需要将请求动态路由到任意一台匹配到的服务器。如下图：



第8步：协议转换

因为有些公司在使用微服务，微服务间使用了 RPC协议，所以在API网关，需要把HTTP协议转换成对应的RPC协议。如下图：



当后端服务处理完请求后会将响应返回给 API网关，网关会将响应转换回HTTP协议，并将响应返回给客户端。API网关还 提供其他关键服务。例如，API网关应跟踪错误，提供断路功能以防止服务过载。API 网关还应该提供日志记录、监控和分析服务，以实现操作可观察性。

监视（Monitoring）和日志记录（Logging）是在API网关和系统中实现可观察性和故障排查的关键组成部分。它们帮助开发人员和系统管理员实时跟踪API的使用情况、性能指标和错误信息，从而更好地了解系统的健康状况，快速发现并解决问题。下面详细解释这两个概念：

监视（Monitoring）是指对API网关和系统中的各种指标和性能数据进行实时收集、分析和展示的过程。通过监视，我们可以了解系统的运行状况、负载情况以及资源使用情况，以便及时采取措施预防或解决潜在的问题。 常见的监视指标包括：

* 请求量和响应时间：跟踪API网关收到的请求数量以及处理请求所花费的时间。
* 错误率：记录API网关处理请求时发生错误的次数，包括HTTP错误码和自定义错误。

* 系统资源使用率：监控CPU、内存、磁盘和网络的使用情况，以确保系统资源充足且没有资源瓶颈。
* 请求队列长度：跟踪等待处理的请求队列的长度，以防止请求堆积导致性能下降。
* 流量趋势：了解API的请求流量趋势，帮助预测系统的负载情况。

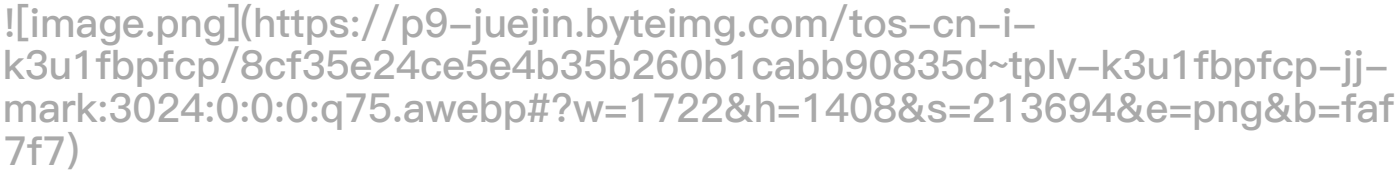
监视可以通过各种监控工具和服务实现，例如Prometheus、Grafana、DataDog等。监视的结果可以以图表、仪表板或警报的形式展示，让开发人员和运维团队能够实时了解系统的状态，做出相应的优化和调整。

日志记录（Logging）是指在API网关和系统中记录关键事件、状态和错误信息的过程。日志记录是一种用于跟踪和调试的重要工具，可以在出现问题时提供有价值的信息，帮助开发人员快速定位问题并进行故障排查。 常见的日志记录内容包括：

* 请求和响应日志：记录API网关收到的每个请求以及对应的响应，包括请求头、请求体、响应码、响应内容等。
* 错误日志：记录API网关处理请求时发生的错误，包括异常、HTTP错误码等。
* 安全日志：记录与安全相关的事件，如认证失败、授权拒绝等。
* 性能日志：记录请求的处理时间、资源使用情况等性能指标。
* 跟踪日志：在多个微服务或后端服务之间，记录请求的追踪信息，帮助跟踪请求的路径和处理过程。

日志可以存储在本地文件系统、数据库中，也可以通过日志聚合工具（如ELK Stack：Elasticsearch、Logstash、Kibana）进行集中管理和分析。日志记录不仅在故障排查时有用，还可以帮助分析用户行为、监控安全风险等。

综合来说，监视和日志记录是在API网关和系统中实现可观察性和故障排查的重要手段。通过监视和日志记录，我们可以及时发现潜在的问题，优化系统性能，并提供更好的用户体验。



总结
==

本文通过分析客户端向服务器发送一个HTTP请求的经典流程，讲解了网关及其主要作用，因为每个公司的业务不一样，所以上述过程也会有差异，另外，因为每个公司基础服务的完善程度不一样，所以对网关的实现不一样。

比如：有些小公司因为业务流量小，直接使用了Nginx充当了网关，然后购买了一些云安全服务，而有些业务体量大的公司，需要单独开发API网关然后集群部署。

总之，网关就是后端服务器的一座保护伞，它对于来自客户端的请求，起到了屏障作用。

交流学习
=====

最后，把猿哥的座右铭送给你：投资自己才是最大的财富。 如果你觉得本文章对你有帮助，点赞，收藏不迷路，为你呈现更多的硬核文章。

原文链接: https://juejin.cn/post/7366874332956065802