24Kubernetes 系列（二十）Kubernetes Gateway API - Server 网络功能进化

Kubernetes 系列（二十）Kubernetes Gateway API - Server 网络功能进化

在本文中，我们将讨论Kubernetes Gateway API 以及它是如何进化 Kubernetes 服务网络的。我们将简要介绍网关API的设计目标，以及它如何改进当前的服务网络标准(如Ingress)。

后端

Service Networking 是 Kubernetes 的一部分，包括在网络上暴露或发布你的 pods，以便其他客户端可以访问它们。Service networking 提供了多种抽象，通过在集群中提供简单的操作方式，或者通过更复杂的方式将 Kubernetes 应用程序公开给运行在集群之外甚至公共互联网上的客户端，从而帮助公开 pods。像Servcie Load Balancer 和 Ingress 这样的 Service Networking 资源通常用于在网络上公开Kubernetes内部运行的应用程序。

Ingress API主要用于使用简单的声明性语法公开 HTTP 应用程序的目标，Ingress 控制器配置底层负载均衡器并完成 Ingress 代理。

Ingress API是为简单的使用场景设计的，支持一些基本的HTTP路由语义:

• HTTP host 匹配

• HTTP path 匹配

• TLS termination

• 路由到 Service 后端

就灵活性而言，Ingress资源并没有提供很多方法来发展这个API以支持更高级的功能。因此，对于高级用例和对新的负载均衡特性日益增长的需求，各个供应商通过使用特定于供应商的 annotations 来处理这些请求。尽管 annotations 确实完成了这项工作，但它造成了不同实现之间的不一致、糟糕的用户体验和安全问题，因为 annotations 可以自由地形成字符串，因此没有验证，而且从一个供应商迁移到另一个供应商变得困难。

Service 资源在使用 Ingress 资源时遇到了几乎类似的问题，它也得到了很多自定义 annotations ，并且由于涉及Kubernetes 不同领域的特性(如负载均衡等)而变得臃肿。

这些问题已经存在了一段时间，并导致现有的 Service networking 资源变得足够复杂，并使其管理和发展变得有点困难。在2019年圣地亚哥KubeCon大会上，一群人聚在一起讨论这些问题，并提出了一种新的API的想法，可以用来解决这些问题。Kubernetes Gateway API 旨在为这些问题提供解决方案，并发展Kubernetes 的 Service networking 领域。

什么是 Kubernetes gateway api?

Kubernetes Gateway API是一个由 SIG-NETWORK 社区管理的开源项目，是一个规范或标准，这意味着支持它的项目和公司必须遵守它。这促进了可移植性和可重用性，因为许多不同的实现具有相同的用户界面。到目前为止，已经有很多像谷歌，RedHat, VMWare, Kong, Traefiklabs等对Gateway API 做出贡献，并且这个列表还在继续增长。这些贡献者和许多其他人也提供了网关API的实现，称为网关控制器。在写这篇文章的时候，官方网站上列出了13种网关控制器的实现，我们相信还有更多的还没有发布到官方网站上。

Gateway API借鉴了 ingress 和服务网格社区的经验，提升了kubernetes -native资源，我们使用这些资源来建模服务网络。网关API增加了如下支持:

• HTTP header-based 匹配

• HTTP header manipulation

• 加权流量分割(weighted traffic splitting)

• 流量镜像(traffic mirroring)

• 面向角色的资源模型

• 等等

它还具有内置在API中的可扩展性和扩展点，以便将来增强，并提供以下支持:

• 任意后端 CRD引用(存储桶、函数等)

• 为现有的不同路由资源提供分层API，例如支持gRPC协议和路由语义

• 粒度扩展点用于特定于实现的行为，如负载均衡算法、自定义匹配类型等

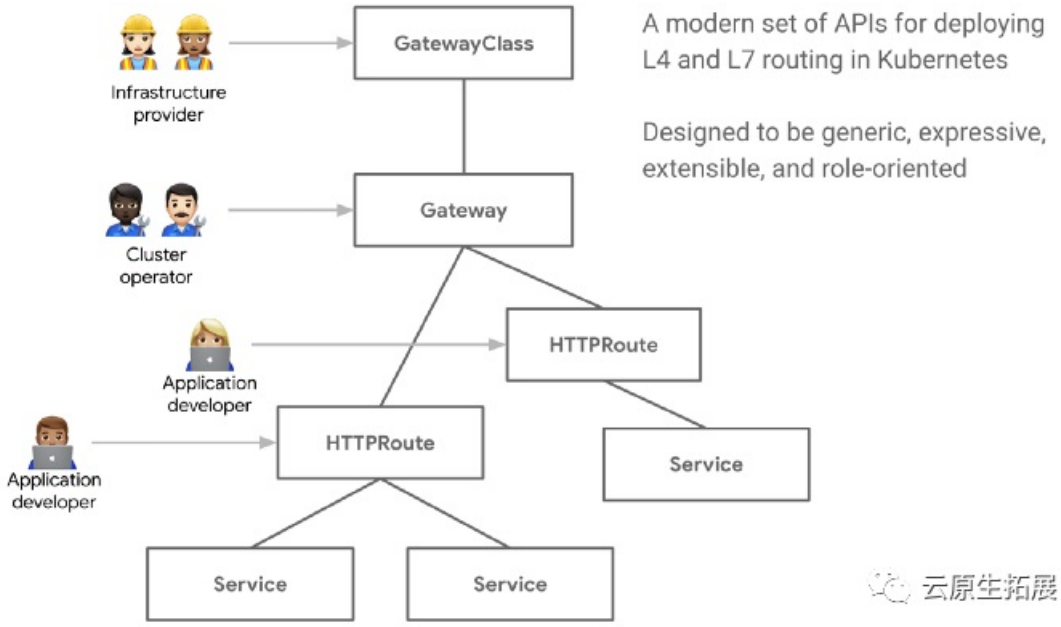

图1:网关API模型

就像 Ingress 控制器代表 Ingress 资源管理网络基础设施一样，Gateway API也有网关控制器来管理底层网络基础设施，比如负载均衡器，它可以是任何东西，从云管理的负载均衡器到集群内的软件代理。每个网关控制器支持一个或多个GatewayClass，其中GatewayClass就像一个模板，它显式地定义了一组共享公共配置和行为的网关。每个GatewayClass将由单个控制器处理，尽管控制器可以处理多个GatewayClass。

网关是从GatewayClass创建的，它们模拟处理流量的实际网络基础设施，就像实际的负载均衡器一样。网关描述如何将流量转换为集群内的服务。也就是说，它定义了一个请求，请求将不知道Kubernetes的地方的流量转换到知道Kubernetes的地方。例如，流量通过云负载均衡器、集群内代理或外部硬件负载均衡器发送到Kubernetes服务。网关被设计成抽象的，因此它们可以为执行路由的许多不同类型的数据平面建模。

然后是 Route Resources ，它定义了将请求从网关映射到Kubernetes服务的特定协议规则。从v1alpha2开始，API中包含了四种Route资源类型，如 HTTPRoute ， TLSRoute ， TCPRoute，和UDPRoute 。特定于实现的自定义路由类型被鼓励用于其他协议。未来可能会在API中添加新的路由类型。

如图所示，我们可以看到网关和HTTP路由资源一起做入口资源作为单个资源所做的事情。这种分离允许不同的角色部署和拥有该资源。

网关API通过Kubernetes网络中面向角色的设计，在为基础设施用户提供灵活性的同时保持基础设施所有者的控制，在分布式灵活性和集中式控制之间取得了平衡。

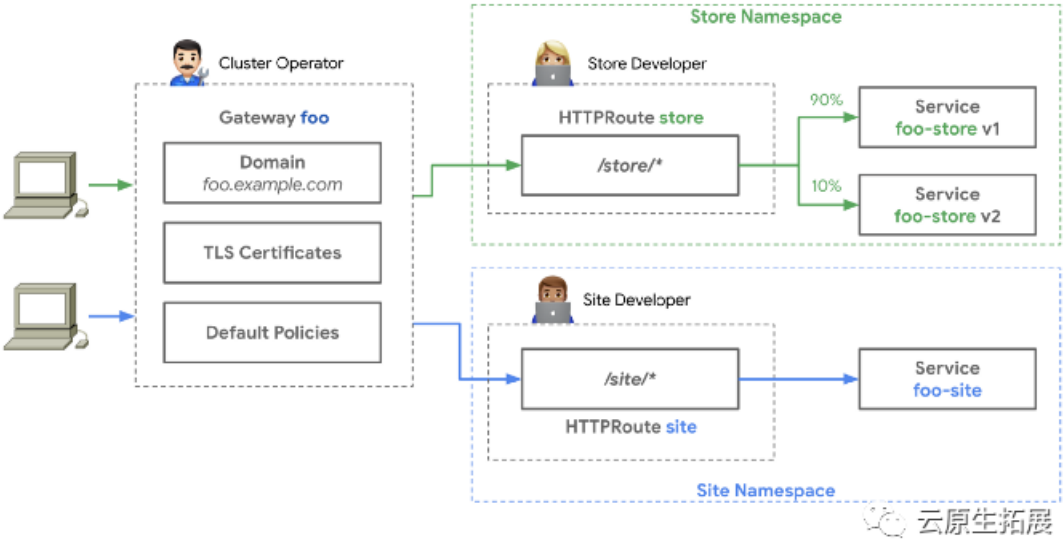

图2:网关API角色

• **基础设施提供商(infra)** 负责集群运行的整体环境。例如，云提供商(AWS、Azure、GCP……)或公司中的PaaS提供商。

• **Cluster operator(ops)** 负责整个集群的管理。它们管理策略、网络访问和应用程序权限。

• **应用程序开发人员(dev)** 负责定义他们的应用程序配置(如超时、请求匹配/过滤器)和服务组合(如到后端路径路由)。

这种角色分离允许许多不同的团队共享相同的Gateway资源，甚至可以跨越命名空间边界。这允许集群操作员在以分布式的方式将路由控制委托给不同的团队时，从本质上控制谁可以访问网关和网关上的策略。

总结

我们已经简要介绍了当前Kubernetes服务网络资源的背景、挑战或复杂性，以及Kubernetes网关API如何提供新的抽象来克服这些挑战。我们还了解到，Kubernetes网关API被设计成面向角色、具有表现力、可扩展和通用的，因此它具有灵活性，可以被不同的组织使用，并在不同的网关控制器集上实现，并为未来的用例保留其核心可移植性。

欢迎关注我的公众号" **云原生拓展** "，原创技术文章第一时间推送。