

Istio 流量管理核心资源

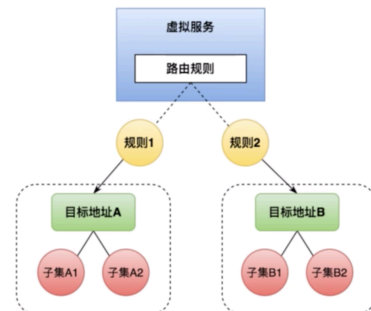
- VirtualService (虚拟服务)
- DestinationRule (目标规则)
- Gateway (网关)
- ServiceEntry (服务入口)

istio中暴露的任何资源都要创建virtualservice，类似于ingress和svc负责负载均衡转发

VirtualService

VirtualService (虚拟服务) :

- 定义路由规则
- 描述满足条件的请求去哪里



destinationrule（与virtualservice配合，将后端的容器分为几组，然后让virtualservice转发到不同的组里面，可以完成灰度发布（如百分之10流量转入到新版本），支持多种负载均衡策略如：随机，权重，和请求数，destinationrule一般用于接收到流量后，怎么帮你转发到后面的pod）

DestinationRule

DestinationRule (目标规则)：定义虚拟服务路由目标地址的真实地址，即子集 (subset)，支持多种负载均衡策略：

- 随机
- 权重
- 最小请求数

#gateway管理这进出流量的转发

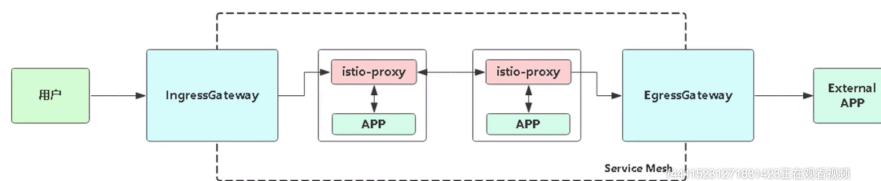
#EgressGateway能管理内部服务访问外部比如数据库之类的流量进行限制，一般情况下不常用。

Gateway



Gateway (网关)：为网格内服务对外访问入口，管理进出网格的流量，根据流入流出方向分为：

- IngressGateway：接收外部访问，并将流量转发到网格内的服务。
- EgressGateway：网格内服务访问外部应用。



#ingress如果想要控制流量比如10分之一流量进入新服务，需要准备至少10个pod，(如果是千分之一不可能准备1000个pod)因为他没发控制流量，而且需要考虑资源情况，而istio-ingressgateway可以完美控制流量层，资源就不需要那么多了。

注意：下面支持的不是4层到7层流量，而是4层和7层

mtls: 双向认证, client和server需要互相认证,一般场景用于银行等金融类企业

Gateway (网关) 与Kubernetes Ingress有什么区别?

Kubernetes Ingress与Gateway都是用于为集群内服务提供访问入口,但Ingress主要功能比较单一,不易于Istio现有流量管理功能集成。

目前Gateway支持的功能:

- 支持L4-L7的负载均衡
- 支持HTTPS和mTLS
- 支持流量镜像、熔断等

#serviceentry 类似于k8s中的endpolint 导入外部服务,进行管理,用的比较少。

ServiceEntry



ServiceEntry (服务入口): 将网格外部服务添加到网格内,像网格内其他服务一样管理。

144116231271801423正在观看22:53

