

Service Mesh Meetup #4 上海站

七牛容器云Service Mesh实践

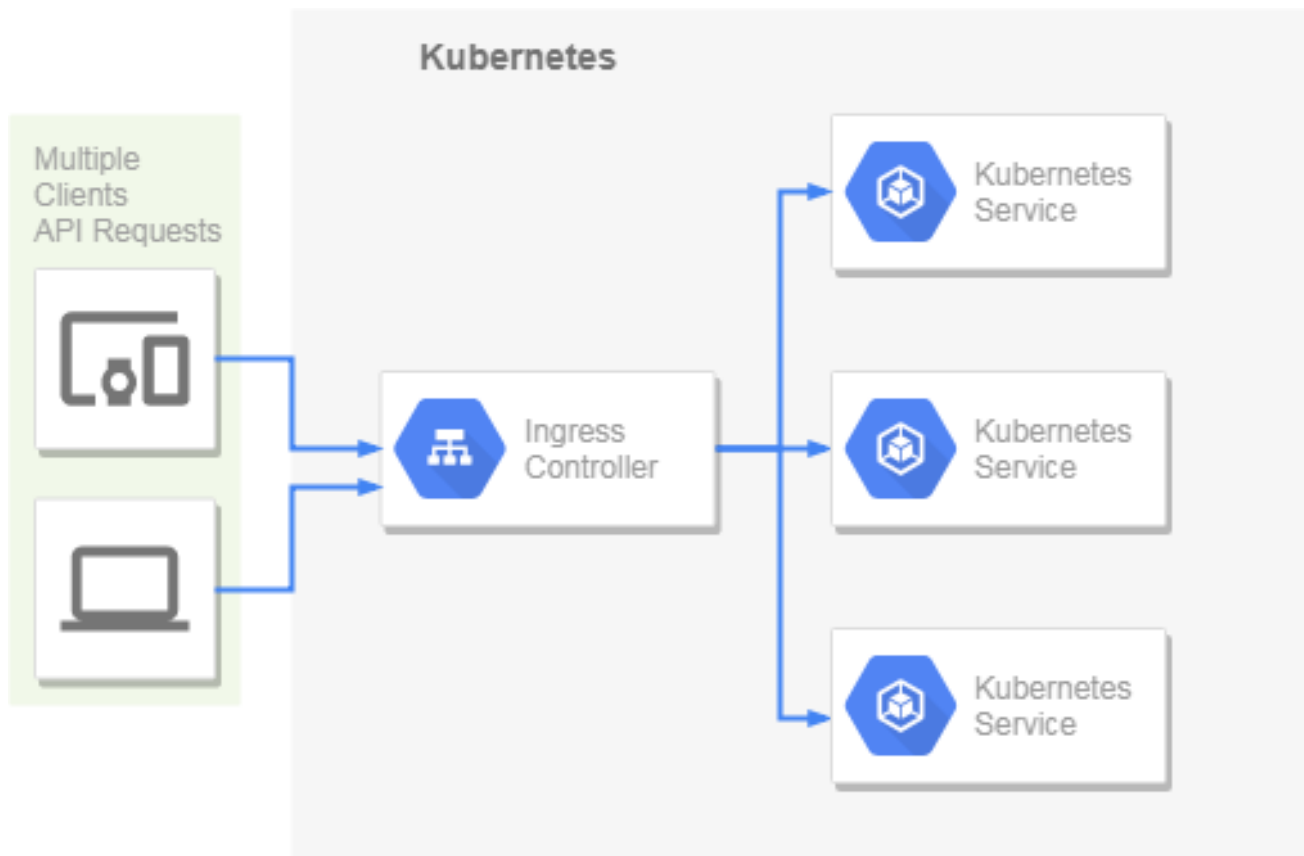
冯玮 七牛容器云架构师

2018.11.25



Ingress Controller

- 流量管理
- 安全管理
- 统一配置
- 反向代理



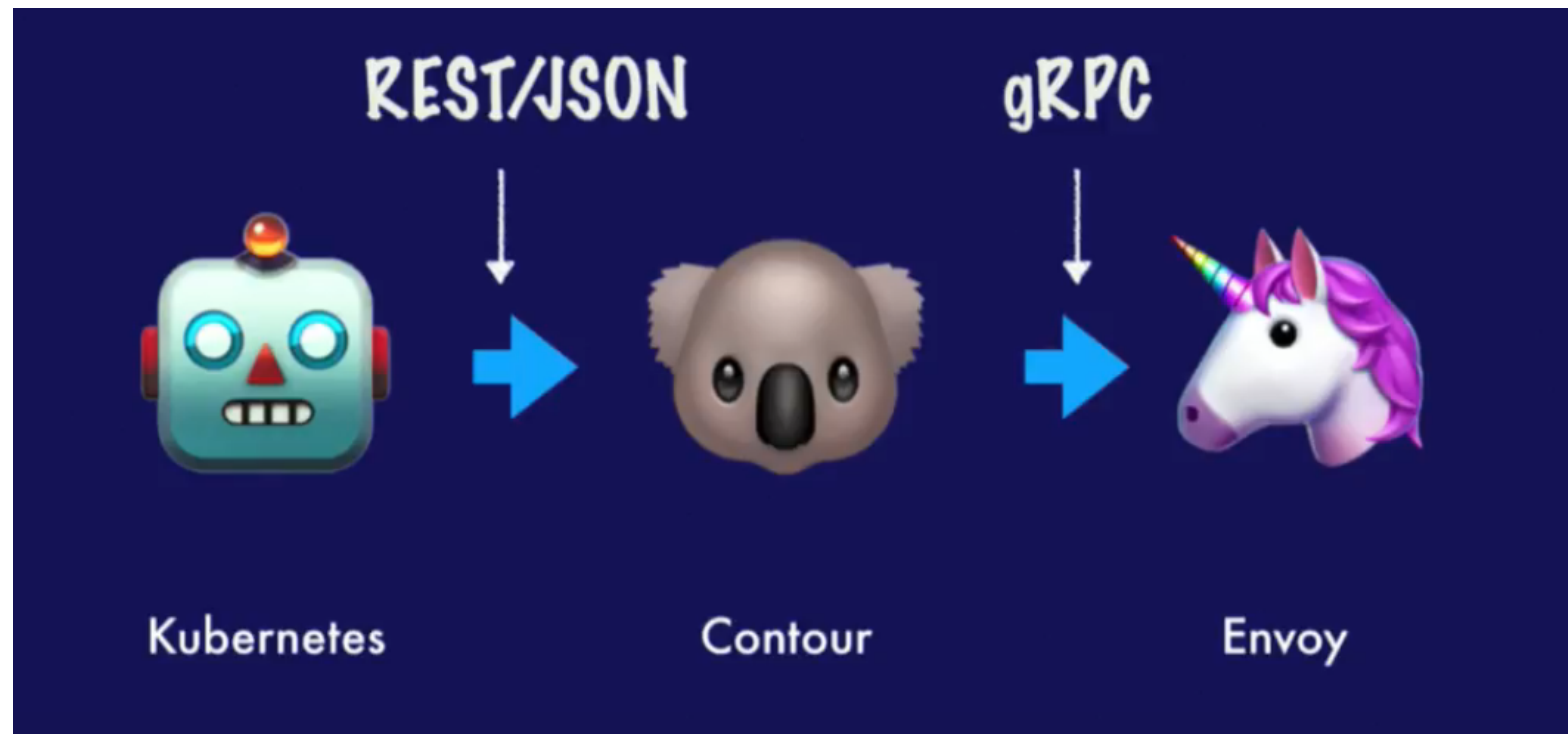
Contour

- 本质上还是Ingress Controller
- Kubernetes深度整合
- Gimbal生态组件



Contour特点

- 基于Envoy
- 协议转换
- 对象翻译
- IngressRouter



Contour的优劣

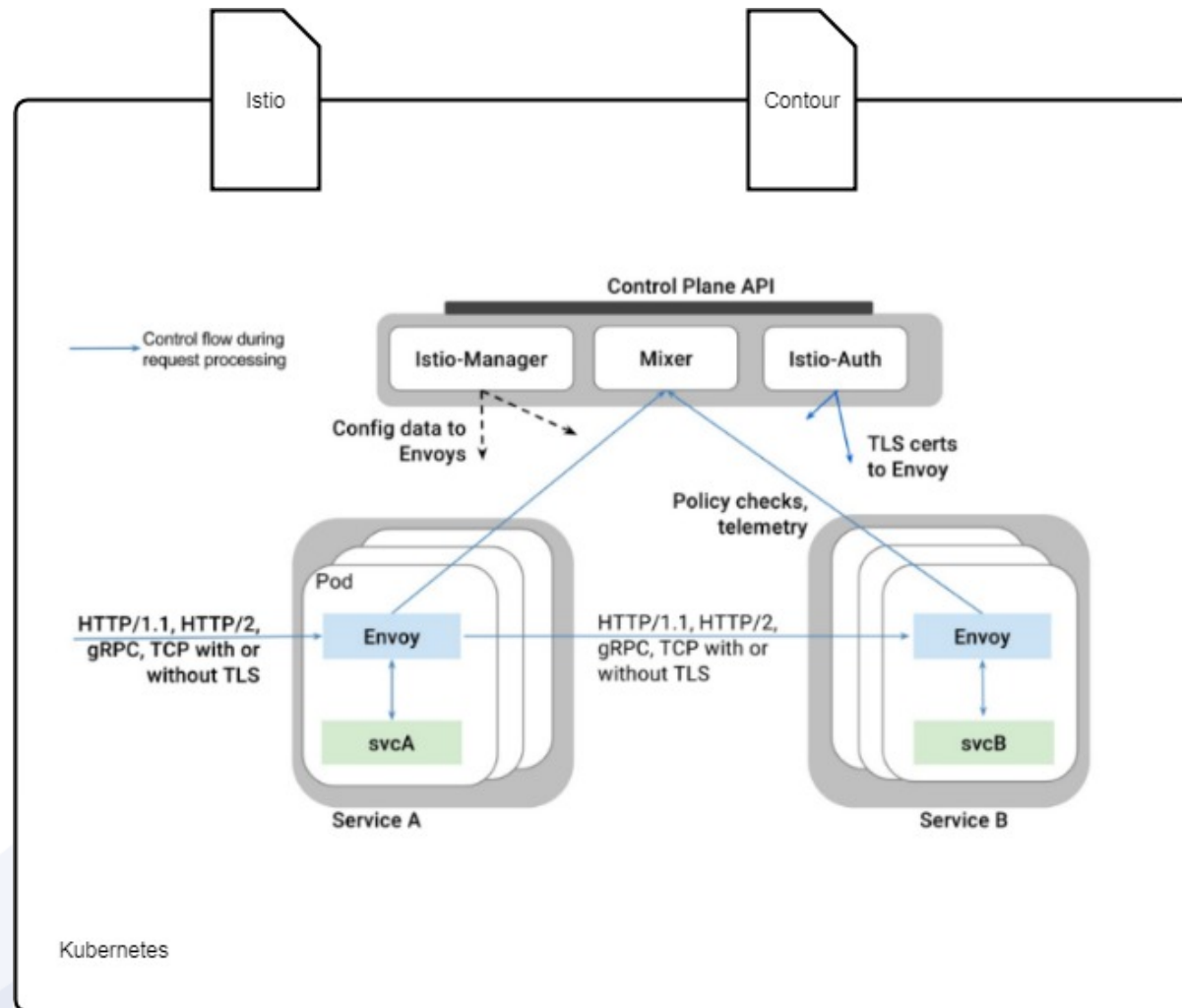
- 优势
 - 兼容Istio生态，融入Service Mesh生态
 - 南北向流量使用Envoy
 - 兼容Kubernetes标准Restful接口
 - 统一的Kubernetes管理接口
 - Gimbal生态
 - 多集群入口流量整合管理
- 劣势
 - 缺少大规模落地案例
 - 功能/非功能仍需加强

Contour & Istio

- 南北向流量
 - API版本共存 (Istio & Kubernetes Ingress)
- 东西向流量
 - Istio支持全量功能
 - Contour支持Tracing能力
- 数据面共存
 - 共用边车组件Envoy

七牛现有Service Mesh体系

- Istio产品化
 - 东西流量产品化
 - 南北流量产品化
 - TLS管理优化
- Contour增强
 - 入口流量管控
 - 跨集群调度
- 发展策略
 - API版本兼容两种方式
 - 数据面优先，控制面按需迭代



七牛容器云Service Mesh发展

- 产品发展
 - 依托容器云PaaS中台
 - 辐射业务线：Spock, Kodo, Dora等
 - 先内部普及踩坑，后私有云能力产品化
- 使用规模
 - 80%以上产品线部署Contour & Istio
 - 涉及K8S集群约20+ 集群
- 功能迭代
 - 先南北，后东西
 - 先原生，后二开
 - 先管控面，后数据面

落地场景—Spock测试平台

- 历史问题

- 分布式系统性能测试问题定位难
- 多版本并发测试
- 故障注入较困难（代码侵入性强）
- 测试环境不稳定，后端Pod频繁重启

- 解决方案

- Contour产品化
- Istio的灰度发布和流量管理
- Istio的Tracing产品化

落地场景—云存储系统

- 历史问题

- 灰度发布
- 预上线系统验证体系
- 系统故障隔离
- 跨集群访问
- 线上问题链路追踪

- 解决方案

- Istio南北流量分流策略产品化
- 基于Istio的QoS产品化
- 跨集群流量调度
- 基于Istio的Tracing产品化

落地场景—大数据产品

- 系统优化之路

- 多版本灰度升级
- 根据流量做横向伸缩
- 分布式系统性能测试问题定位难

- 解决方案

- Istio南北流量分流策略产品化
- 基于Istio的QoS产品化
- 基于Istio的Tracing产品化
- 跨集群流量调度

七牛容器云产品逻辑架构

Kubernetes + Docker + 生态链

私有云

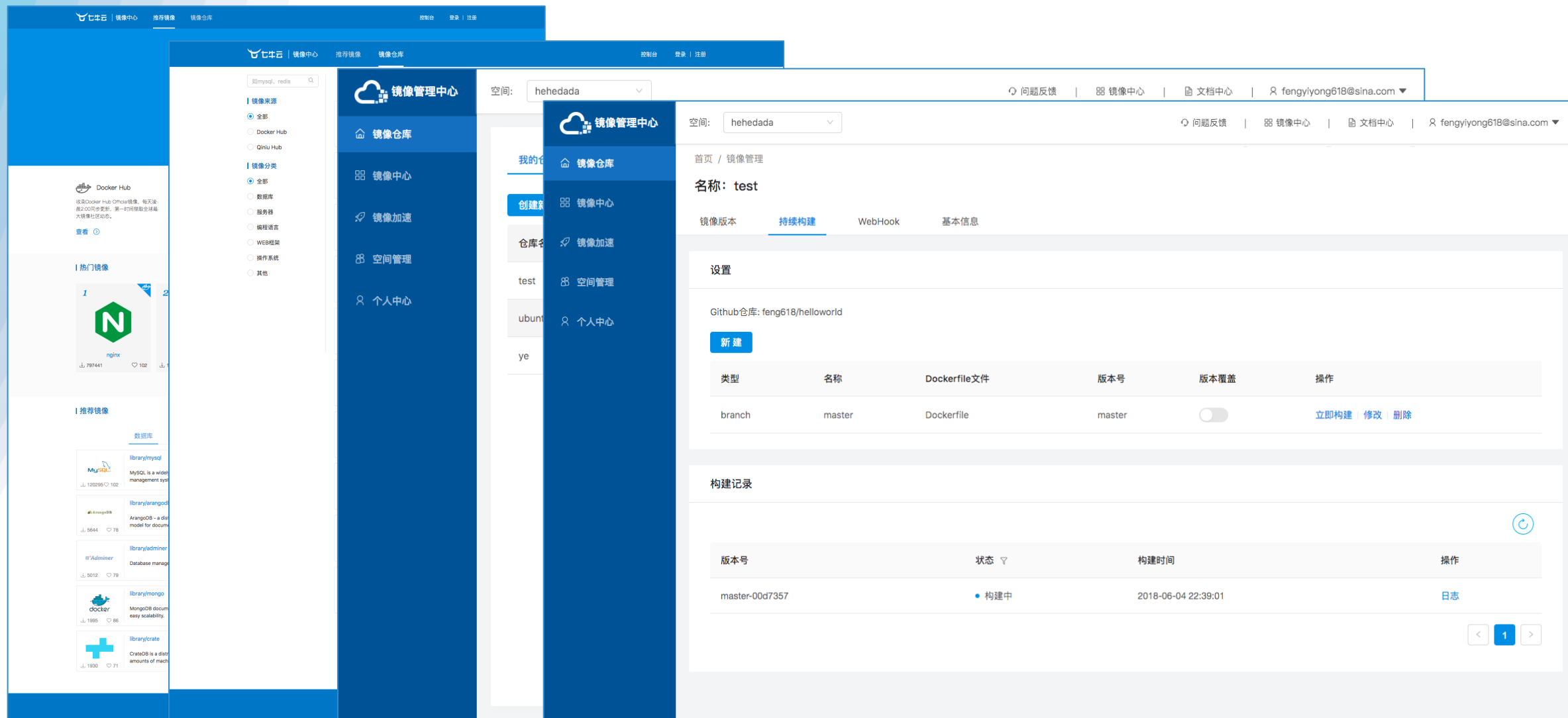
公有云

CI/CD ; 监控告警平台 ; 日志系统 ;
应用管理 ; 多租户支持 ; DevOps工具链

Kubernetes + Docker

IaaS (物理机 , 云主机 + 基础网络+ 存储服务)

七牛容器云产品



未来发展

- 更多功能
 - 故障熔断
 - 故障注入
 - 业务配置动态分发
- 优化升级
 - 性能优化：
 - DPDK + eBPF
 - 系统优化
 - 根据需求裁剪Istio。解耦Envoy和其他组件
 - 裁剪Istio相关CRD，简化系统结构

ServiceMesher公众号



SOFAStack公众号



<http://www.servicemesher.com>

