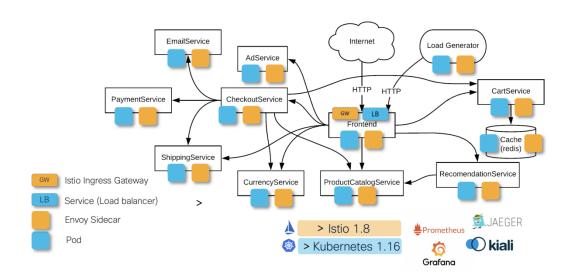
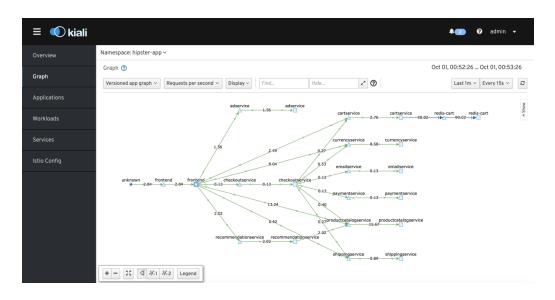
研究基于云原生微服务架构的 web 商城应用 Online Boutique, 部署在 k8s 环境中, 并进行深入的研究。详见 https://github.com/GoogleCloudPlatform/microservices-demo

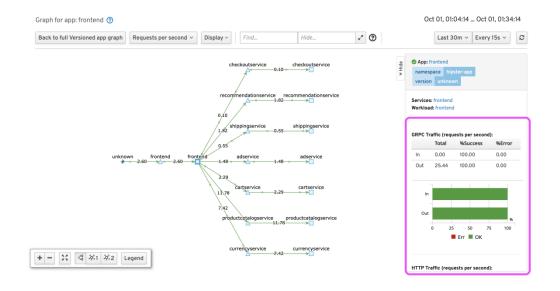
要求:

(1)针对 Online Boutique 在熔断、限流、监控、认证、授权、安全、负载等方面的不足,将其升级到服务网格架构,为微服务启用 Istio 支持。同时,为了启动 Istio 支持,需要为 POD 额外注入一个 SideCar 应用。

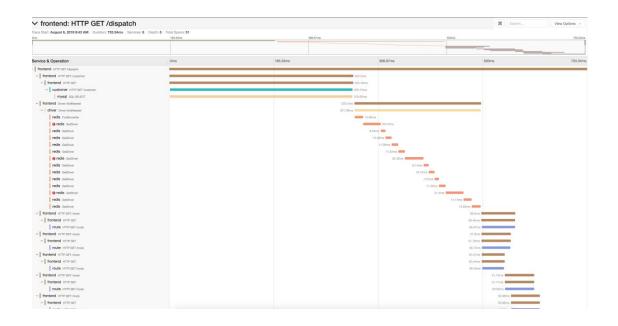


(2) 在 k8s 中为 Istio 配置 Kiali,实现 Istio 服务网格的可视化,为 Online Boutique 项目提供服务拓扑图、全链路跟踪、指标遥测、配置校验、健康检查等功能。在使用 Locust 压力负载的情况下,完成服务网格的各项指标遥测、检测和健康检查。





(3) 在 k8s 中为 Istio 配置 Jaeger,为 Online Boutique 项目提供分布式 调用链追踪系统,用于监控和排查基于微服务的分布式系统问题,包括:分布式上下文传播、分布式事务监控、根因分析、服务依赖关系分析、性能 / 延迟优化等任务。在使用 Locust 压力负载的情况下,完成服务网格中各微服务的性能 /延迟监测和分析。



(4)*在 Kubernetes 中安装和配置 Prometheus、Grafana,完成整个集群、工作负载在使用 Locust 压力负载的情况下的运行情况、健康状态监控。



- (5)*在 Online Boutique 项目中,将其中至少一个非 Java 实现的微服务,改写成 SpringBoot 的微服务,完成整体新项目 Kubernetes 中的部署,重新使用步骤 1-4 中的组件,完成调试、分析和管理任务。
- (6) 总结和整理以上各项技术平台的底层实现机制,以及在 Online Boutique 项目中的配置和使用步骤。