GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

微服务和K8S集成-探索实践

邢海涛
灵雀云微服务首席专家

2017.thegiac.com

微服务对DevOps的挑战

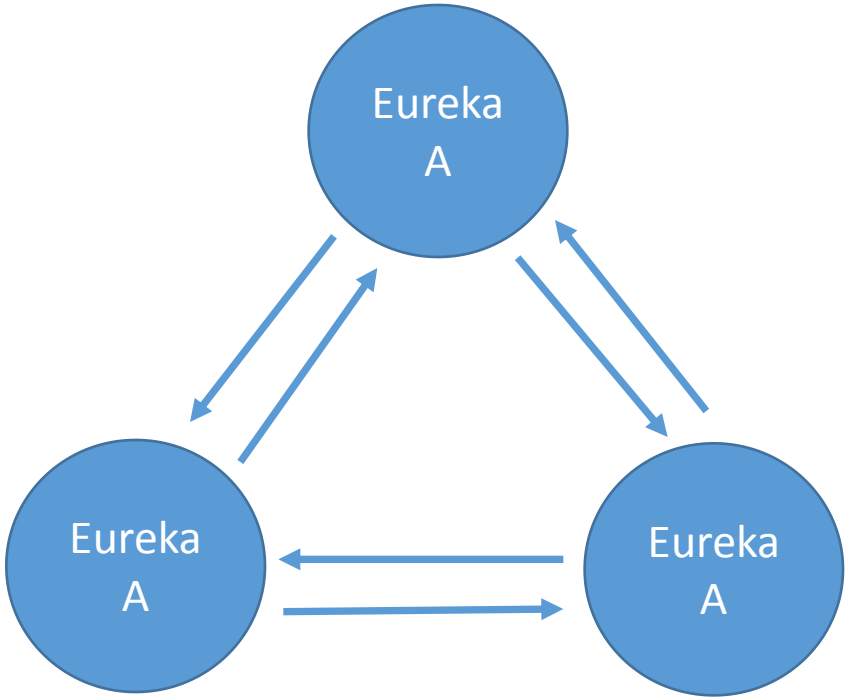
- 快速配置计算资源
- 基本监控
- 持续集成，持续交付，持续部署
- 易于配置存储
- 认证/授权
- 网络的问题
- 服务管理

Kubernetes - 容器编排王者

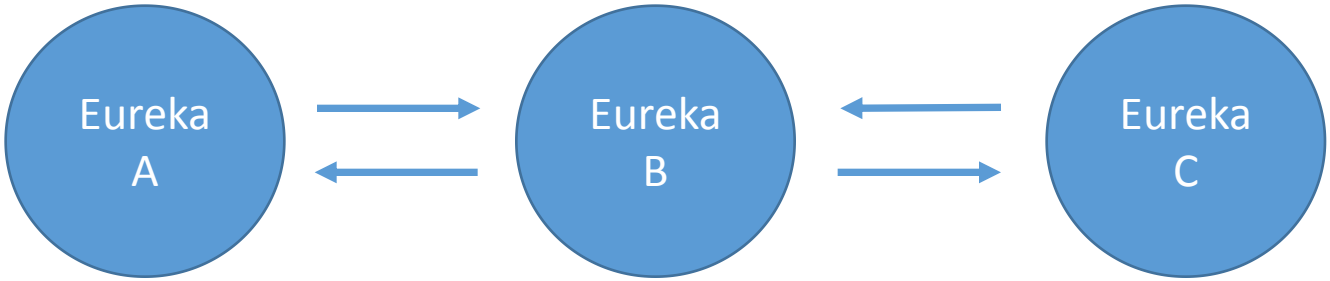
- 资源调度
- 弹性伸缩
- 自动化部署
- 滚动发布，蓝绿发布，灰度发布

服务发现 — Eureka高可用

- 部署三个或更多对等节点
- 注册动作会传递
- 客户端同时向所有Eureka注册



服务发现 — Eureka高可用



节点A和C无法交换数据

服务发现 — Eureka高可用

方案1：K8S Headless + DaemonSets

K8S Headless Service

没有负载均衡

1个服务名标明集群

多个A记录对于此服务名

K8S DaemonSets
(生产环境可选)

1个Node节点1个Pod

所有节点或部分节点

Eureka扩展

读取DNS信息，把1个服务名转化为多个节点的IP地址

服务器端：扩展EurekaClientConfigBean，同时过滤自身IP

客户端：扩展EurekaInstanceConfigBean

服务发现 — Eureka高可用

方案1：K8S Headless + DaemonSets

优点

简洁的配置

K8S的便利

缺点

需要扩展Eureka

服务发现 — Eureka高可用

方案2：K8S StatefulSets

K8S StatefulSets

要求Headless

每个实例对应的网络ID

网络Id列表标明集群

持久的存储卷Volume

Eureka扩展
（可选由于网络Id存在）

读取DNS信息，把1个服务名转化为多个节点的IP地址

服务器端：扩展EurekaClientConfigBean，同时过滤自身IP

服务发现 — Eureka高可用

方案2：K8S StatefulSets

优点

简洁的配置

K8S的便利

缺点

需要扩展Eureka(可选)

配置管理 - ConfigMap

ConfigMap到Pod环境变量

单个ConfigMap数据

多个ConfigMap数据

ConfigMap到存储卷

存储到文件

指定路径和权限

自动更新

服务发现 — No Service Discovery

你满意Spring Cloud?

AP系统(Consul, Eureka) or CP系统(Zookeeper, etcd)?

多语言支持?

只使用DNS?

- 弹性DNS，动态DNS

只使用Kubernetes服务?

- 客户端负载平衡? Ribbon
- Turbine / Hystrix Dashboard

服务发现 — No Service Discovery

方案： K8S + spring-cloud-kubernetes [1]

1. DiscoveryClient for Kubernetes
2. Ribbon discovery in Kubernetes
3. Zipkin discovery in Kubernetes

服务发现 — No Service Discovery

方案： K8S + spring-cloud-kubernetes [1]

优点

K8S提供的便利

多语言微服务

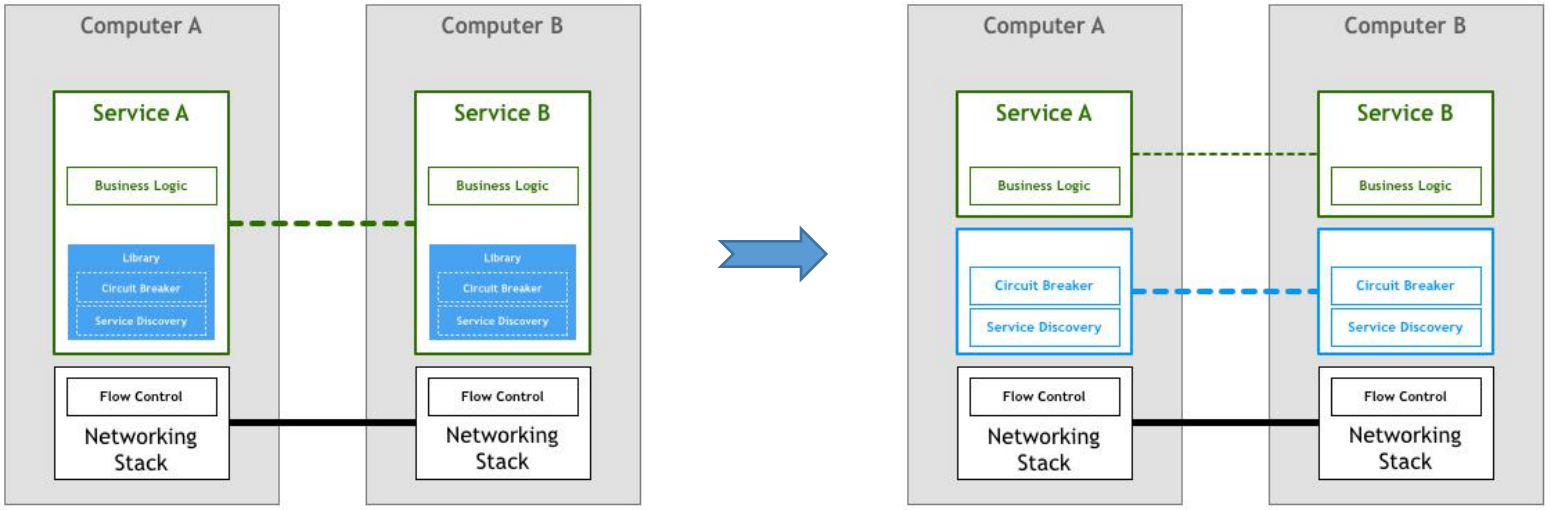
简化原有SpringCloud部署

缺点

代码的K8S依赖

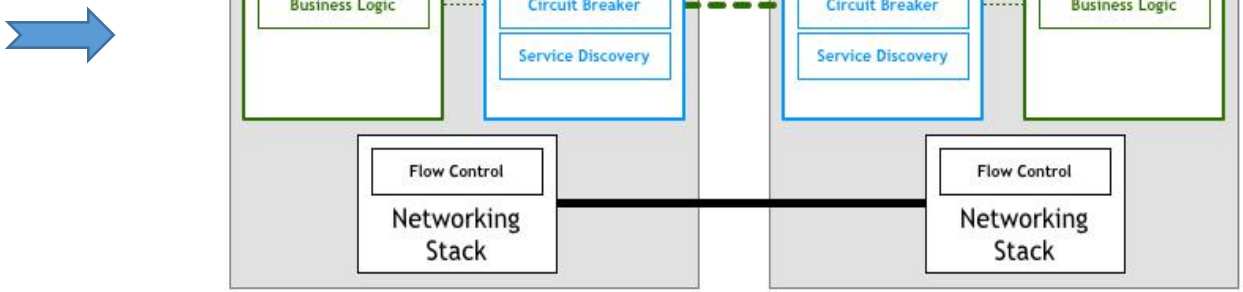
需求转化：从Eureka高可用到K8S高可用

Service Mesh — 演变



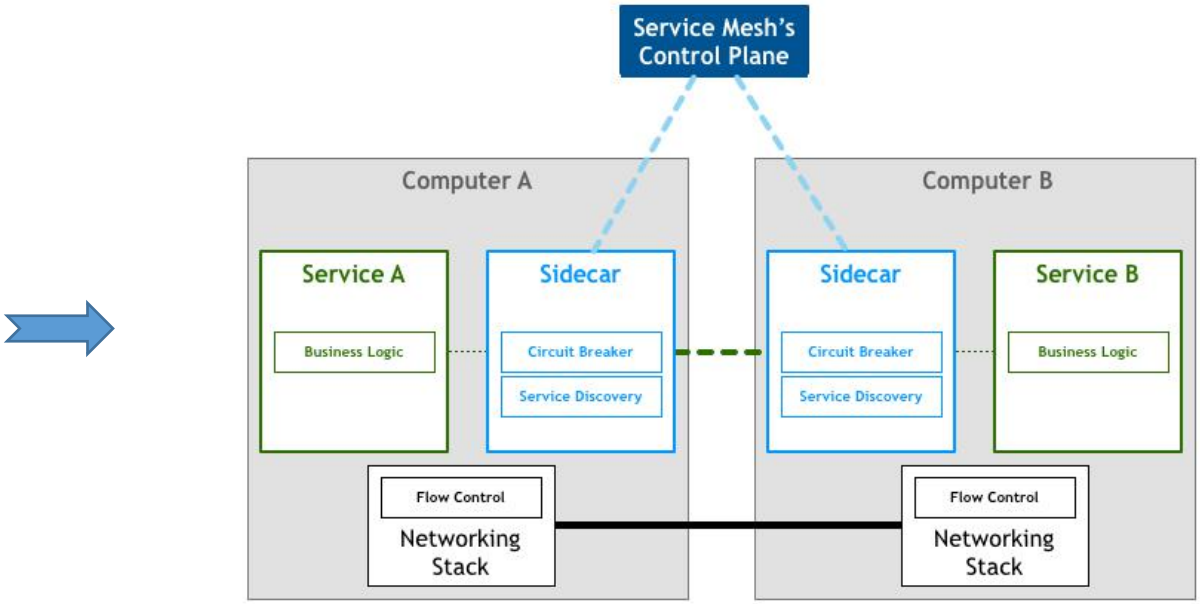
Pattern: Service Mesh [2]

Service Mesh — 演变



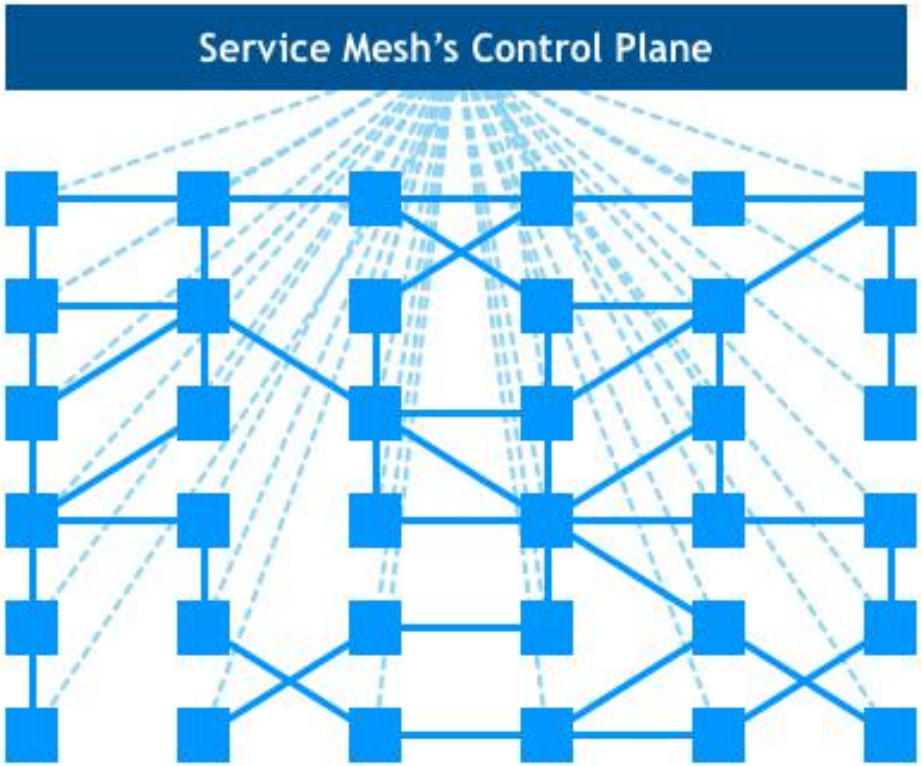
Pattern: Service Mesh [2]

Service Mesh — 演变



Pattern: Service Mesh [2]

Service Mesh — 演变



Pattern: Service Mesh [2]

Service Mesh — 定义

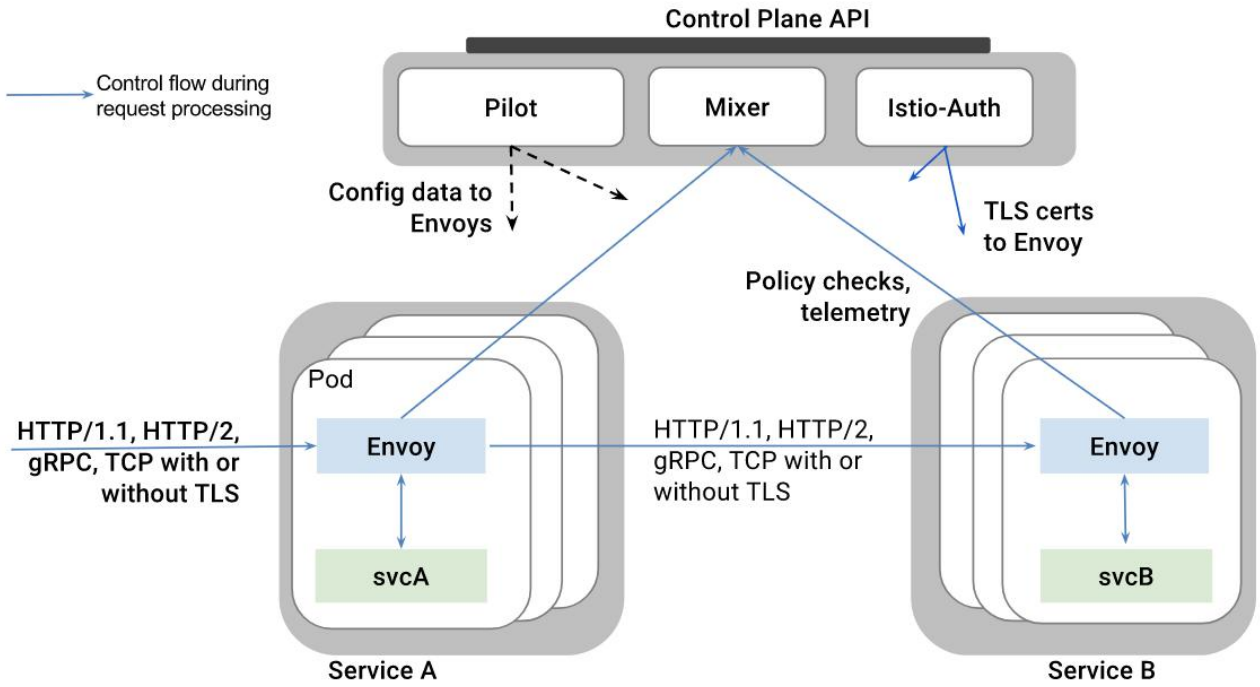
Service Mesh [3]

处理服务到服务通信的专用基础设施层

透过云原生应用程序的复杂拓扑结构，负责可靠地传递请求

实践中，Service Mesh通常被实现为与应用程序代码一起部署的一组轻量级网络代理，应用程序无需知道代理组的存在

Service Mesh — Istio 实现



Istio Architecture

Service Mesh — No Service Discovery & Even Less

方案：K8S + Service Mesh 敬请期待！

Linkerd [4]

使用K8S DaemonSets

K8S Ingress Controller

Istio [4]集成作为控制面板

Conduit [5]

K8S紧密集成

K8S Deployment作为管理单元

Istio [6]

K8S部署

References

- [1] https://github.com/spring-cloud-incubator/spring-cloud-kubernetes
- [2] http://philcalcado.com/2017/08/03/pattern_service_mesh.html
- [3] https://buoyant.io/2017/04/25/whats-a-service-mesh-and-why-do-i-need-one/
- [4] https://linkerd.io
- [5] https://conduit.io
- [6] https://istio.io

GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

扫码关注GIAC公众号

2017.thegiac.com