

K8s-hook

Wen Zhenglin

2018-9-13

概览

1. K8s Service概念
2. Hook机制
3. 遇到的问题
4. 示例K8s源代码编译

K8s Service概念

- 有自己的IP(虚拟IP)
- 用于负载均衡连接后端(具体的容器实例)
- 服务一般不变, 容器变化较多(pod重启, 即ip变化)

没用k8s service

1. 性能的考虑(dns当时处于非常不稳定的状态, iptables)
2. 标准化的考虑(service需要用ingress, nginx的配置需标准化, 目前不支持)
3. 一套独立的nginx和现有的nginx分离, 增加管理成本
4. 从当时看ingress技术尚未成熟

需要我们有自己的类service
提供服务访问

What if pod (ip) change?

问题

Watch假死 (<http://issue.qianbao-inc.com/Cloud/req/issues/43>)

可能的原因: Apiserver down导致Watch假死

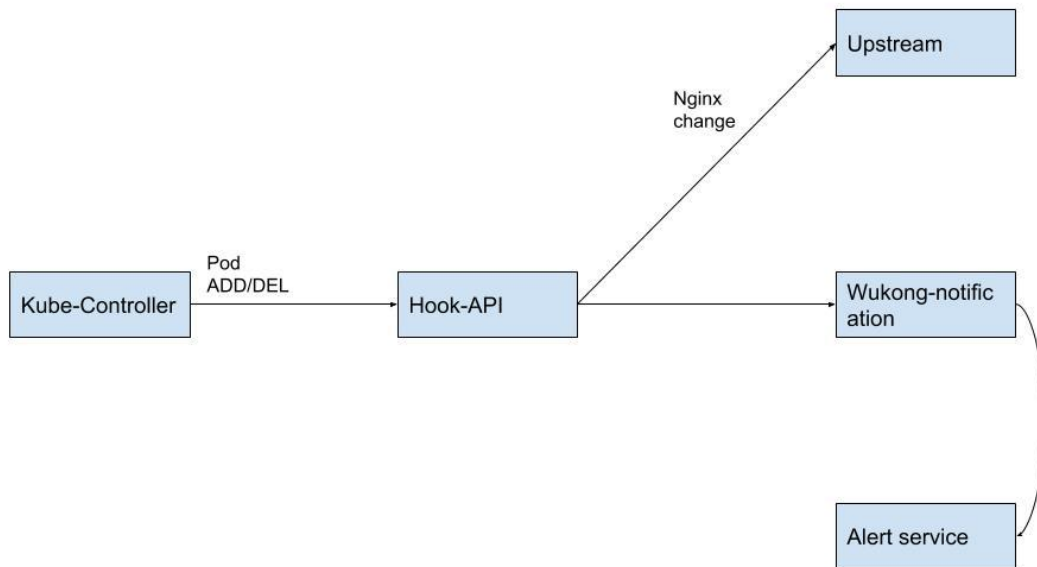
问题的解决:

1. 现有基础上加入重连机制(已实现)
2. 修改K8s组件, 加入hook机制(主动通知外部作出变化)

介绍k8s-hook

1. 解决watch假死问题
2. 添加upstream时提供服务状态检测(解决初始时服务不可用问题)
3. 更快的响应机制(非轮询机制)

k8s-hook



源代码

1. <http://issue.qianbao-inc.com/Cloud/kubernetes/src/branch/AddHook/pkg/controller/endpoint>
2. <http://issue.qianbao-inc.com/Cloud/hook-api>

Source code explain

1. K8s source code
2. Hook source code

遇到过的问题

1. 模式匹配(从众多信息中抽取模式)
2. 删除不够快(模式变更)
3. 添加upstream前检测
4. 部署方式(自动化)

1. 模式匹配

1. 基于阶段类型(add and delete endpoint)
2. 基于IP和端口组合
3. 基于字段(reduce the event to only one)

2. 删除不够快(模式变更)

从30秒到1秒删除

1. 改变了新的模式匹配解决

3. 添加upstream前检测

范围内的多次重复检测

1. 通过项目中心提供的检测参数进行检测
2. 简单的tcp端口检测

与服务治理结合

4. 部署方式(自动化)

1. Build和上传 (./build.sh)
2. 下载和重启 (./update.sh)

总结

介绍了K8s服务与我们系统之间的交互，服务的工作机制和具体hook的实现方法。

介绍了其中遇到的问题及如何解决的。

Thank you!

Bonus

hook-api与SVC-d服务治理的关系

1. Hook-api 处理容器启动和停止
2. SVC-d处理容器运行时