

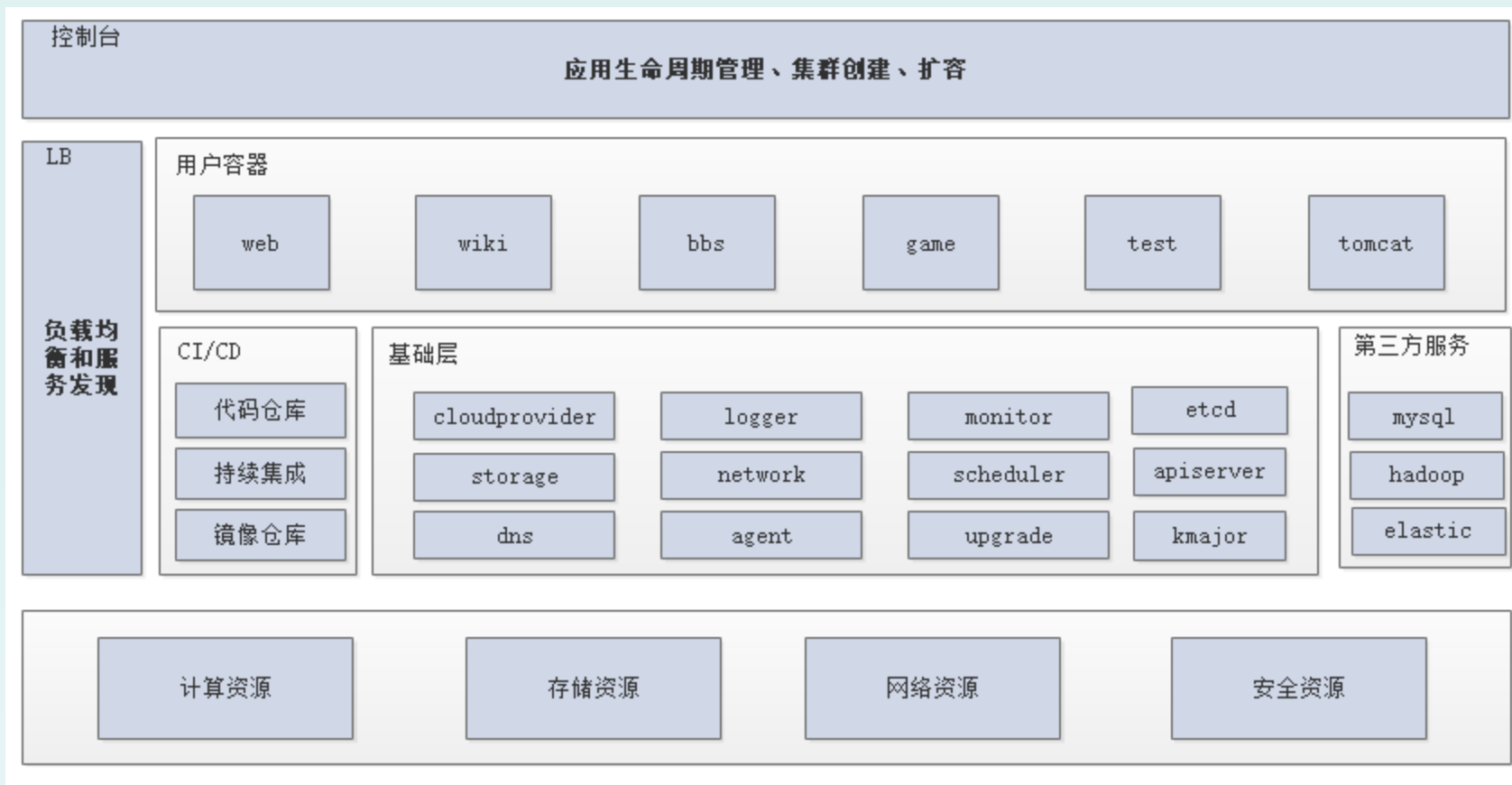
k8s服务发现与负载均衡

吴利斌 2017/12/12

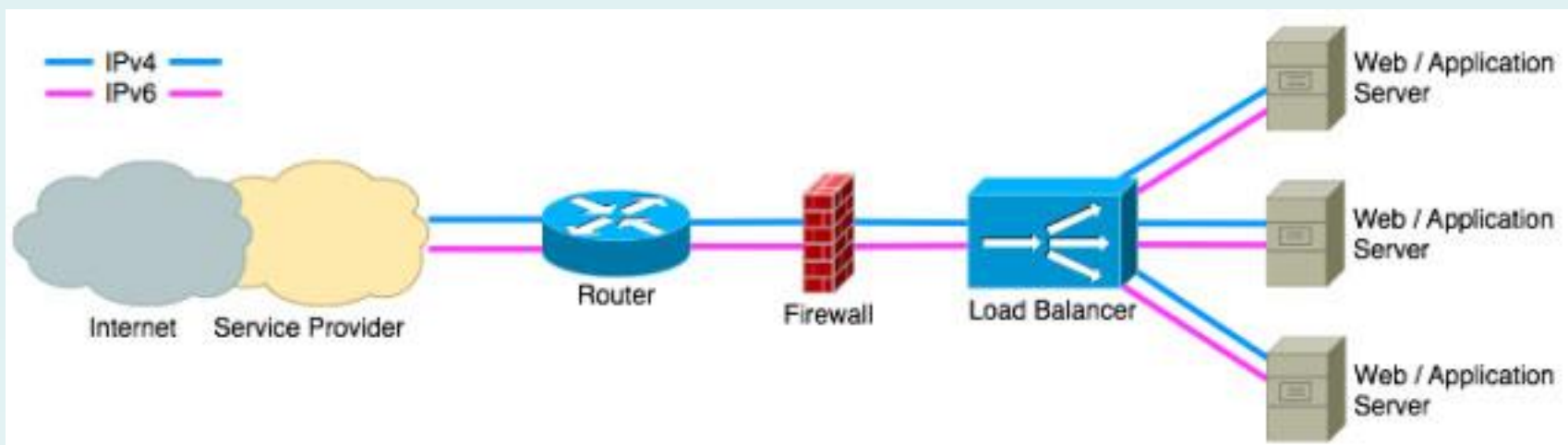
目录

- 1、什么是负载均衡？
- 2、容器化环境下有什么问题？
- 3、k8s怎么解决？
- 4、还有什么问题？

整体架构



负载均衡

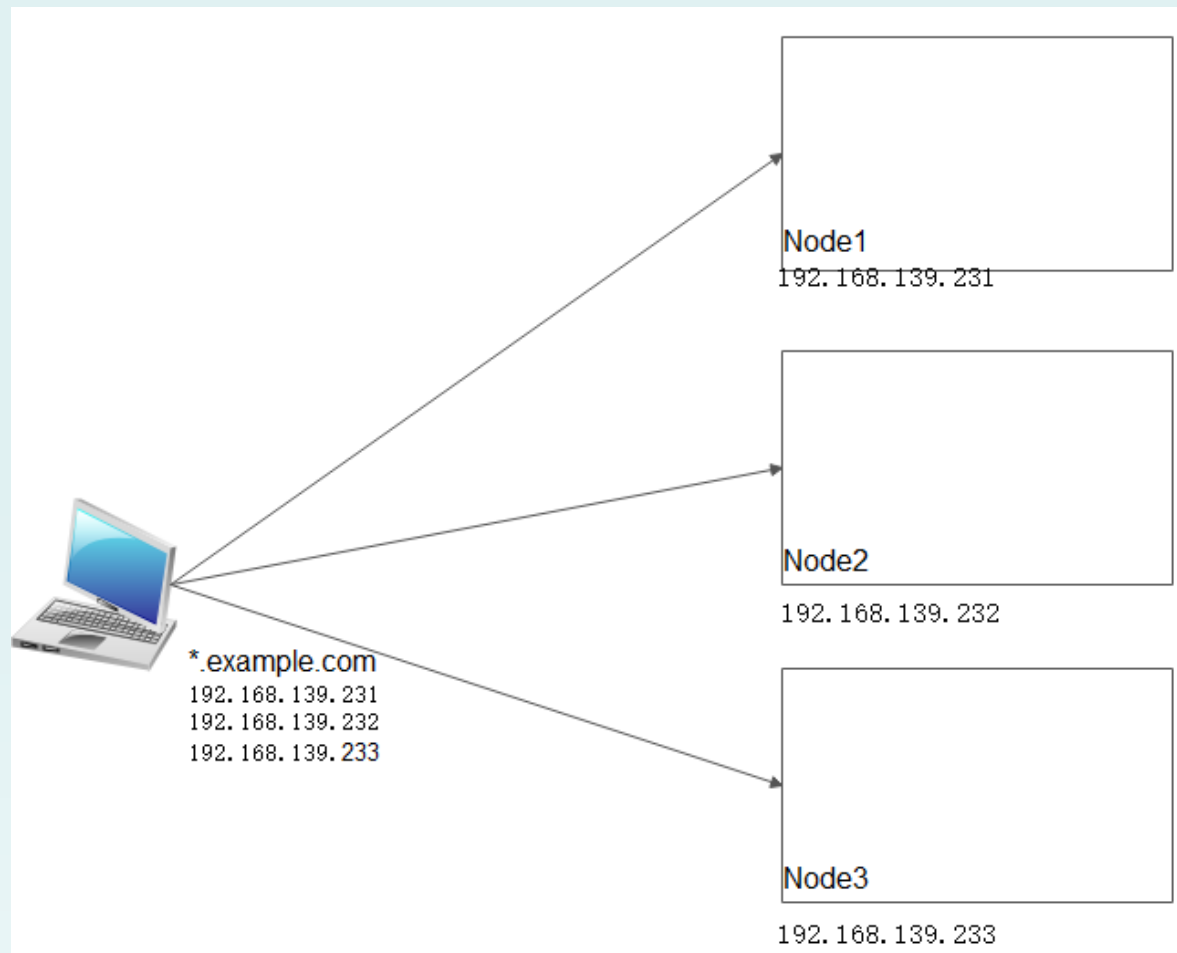


客户端负载均衡

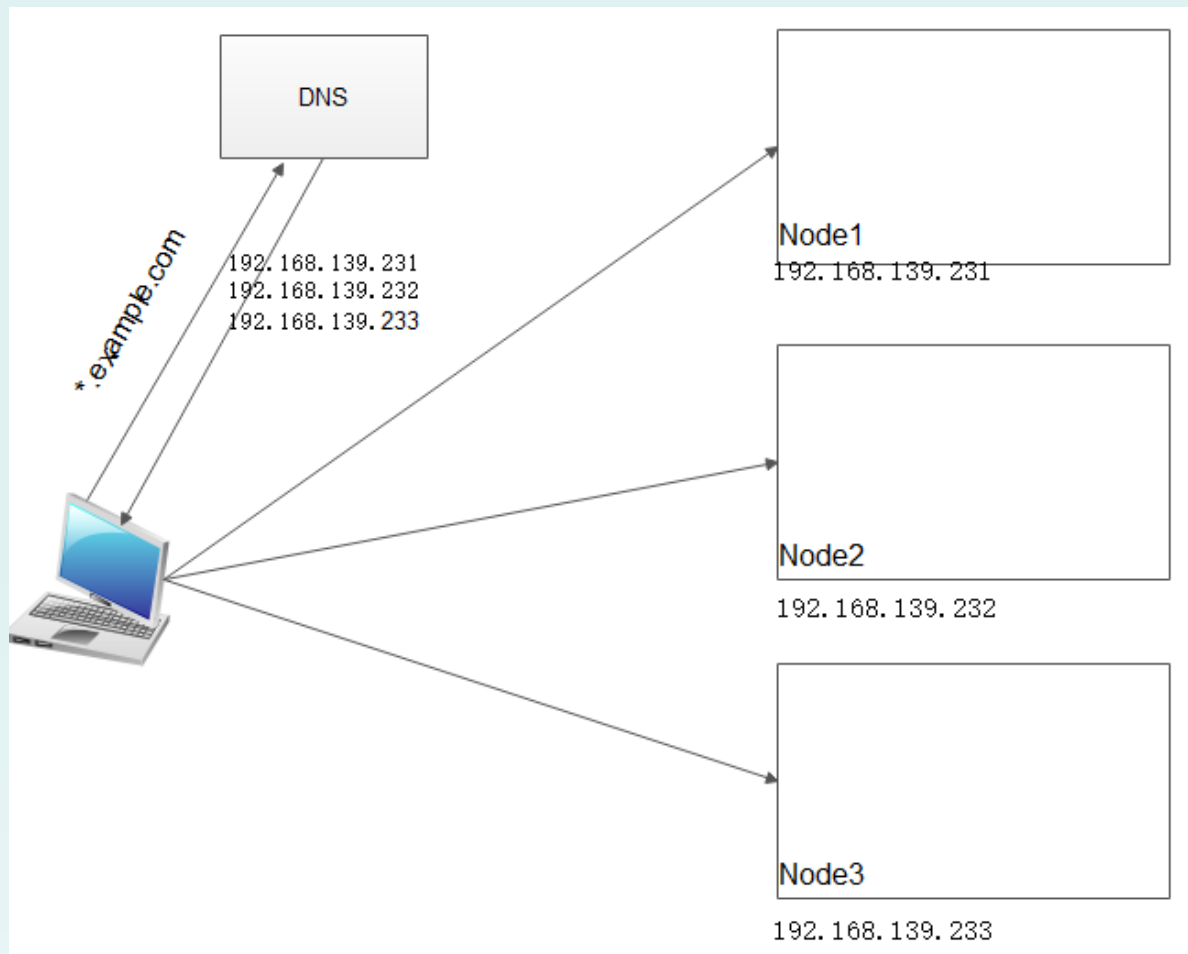
1、客户端指定一组服务端地址

2、客户端通过一定负载均衡策略去访问服务端

缺点：需要客户端支持，普通web服务器不支持



通过DNS来做



1、DNS查询某个域名的A纪录会返回该域名对应的一个或多个IP地址

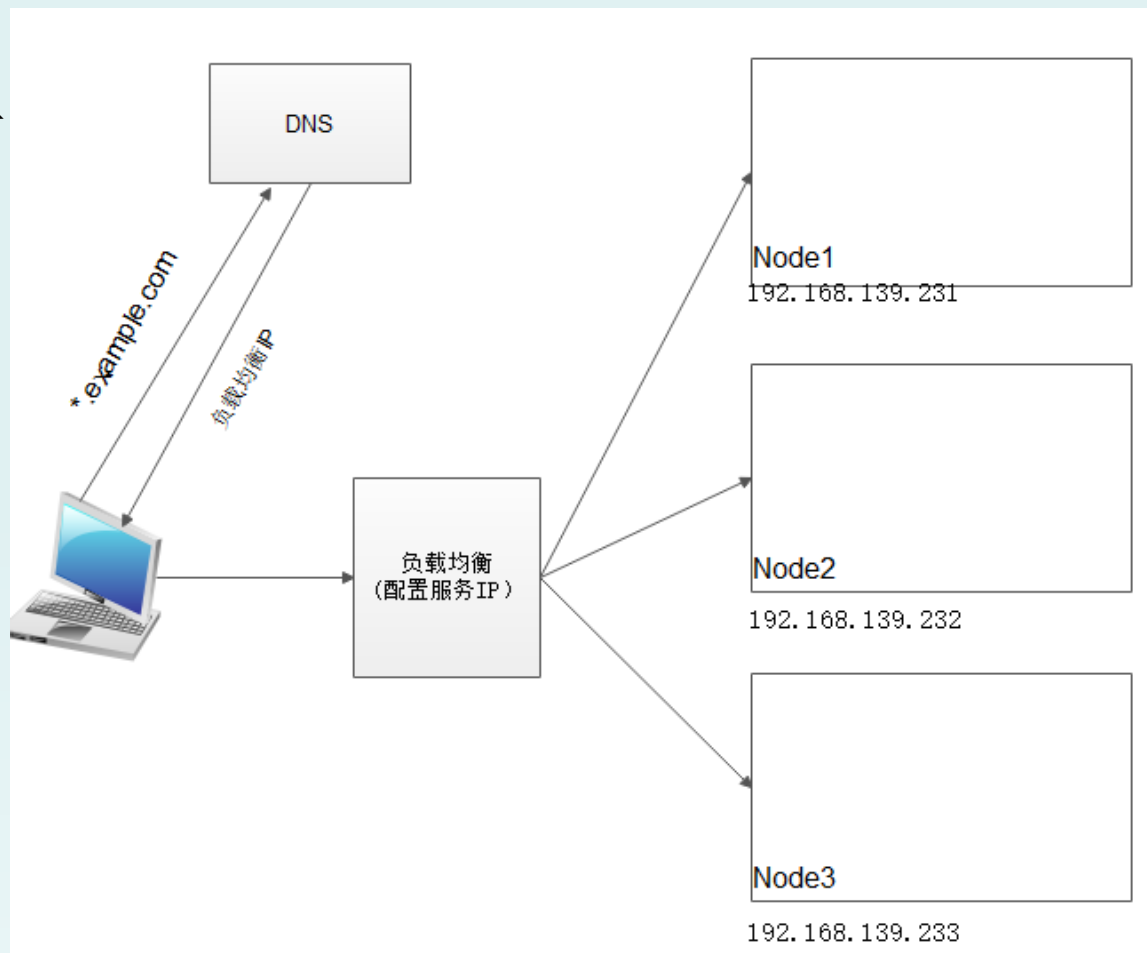
2、DNS可以根据用户请求，动态的返回某域名的A纪录

缺点：每次加服务都要改dns，生效慢；负载不够均衡，效果差

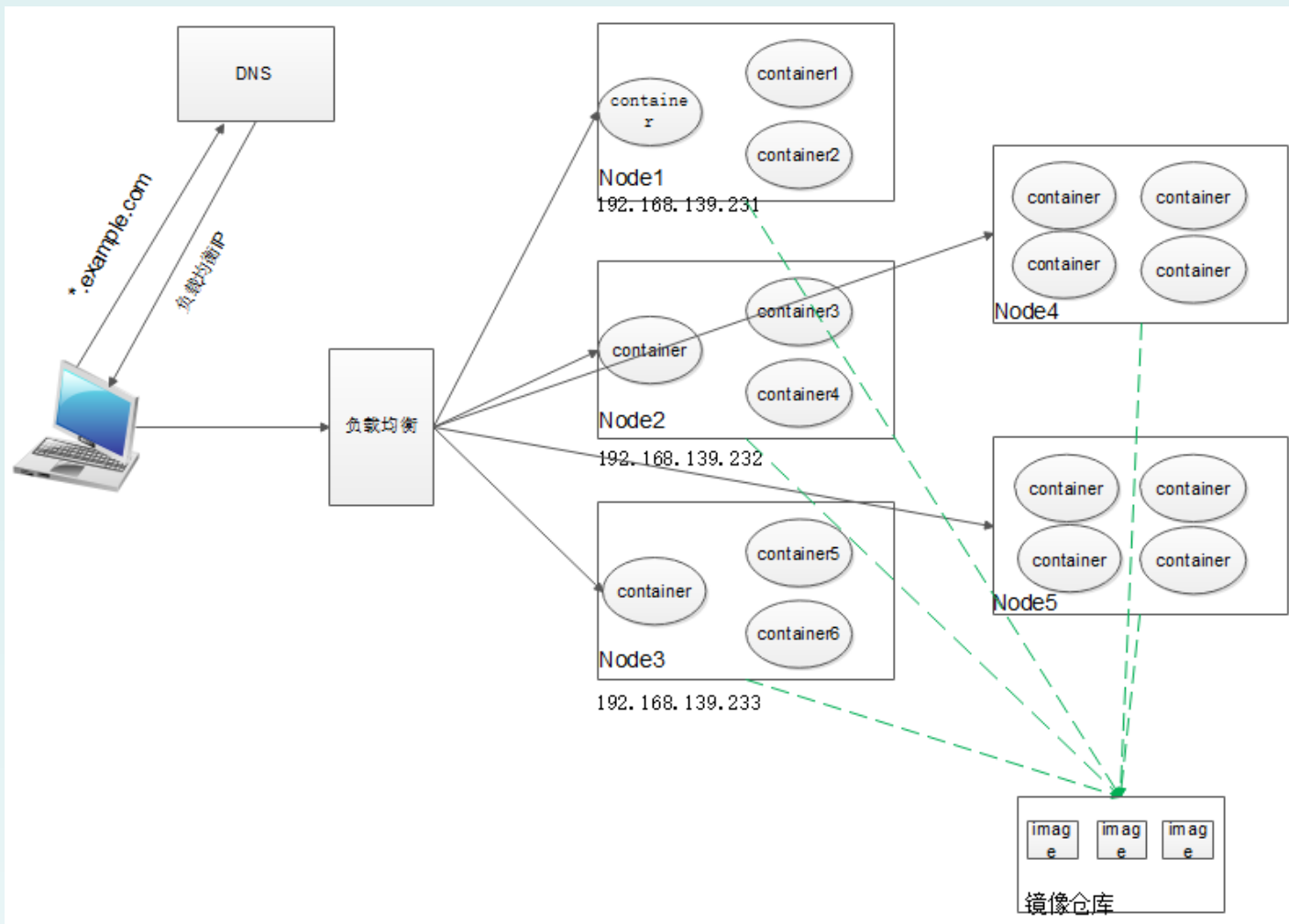
加前置服务均衡器

优点:

- 1、扩容后，只需改前置即可
- 2、负载策略可以自己选
- 3、对前后端都是透明的



容器化环境下



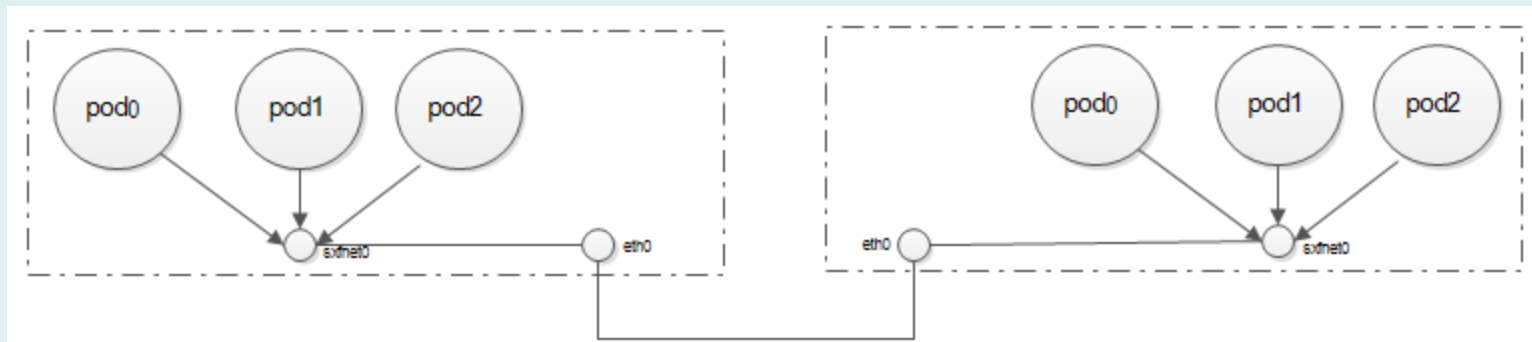
问题

- 1、应用的容器化和微服务化，导致应用粒度变小，被更加快速地创建和删除，这样原先的固定IP访问方式就失效了
- 2、IP频繁发生变动，动态应用部署无法预知容器的IP地址，client端如何发现server端的访问端点？
- 3、2个问题需要重新考虑：服务发现和负载均衡。

K8s怎么解决

- 1、主动注册机制（服务发现）
- 2、service
- 3、ingress

k8s网络模型



1、每个Pod都拥有一个独立的IP地址（IPper Pod），而且假定所有的pod都在一个可以直接连通的、扁平的网络空间中。

2、用户不需要额外考虑如何建立Pod之间的连接，也不需要考虑将容器端口映射到主机端口等问题。

3、网络要求：

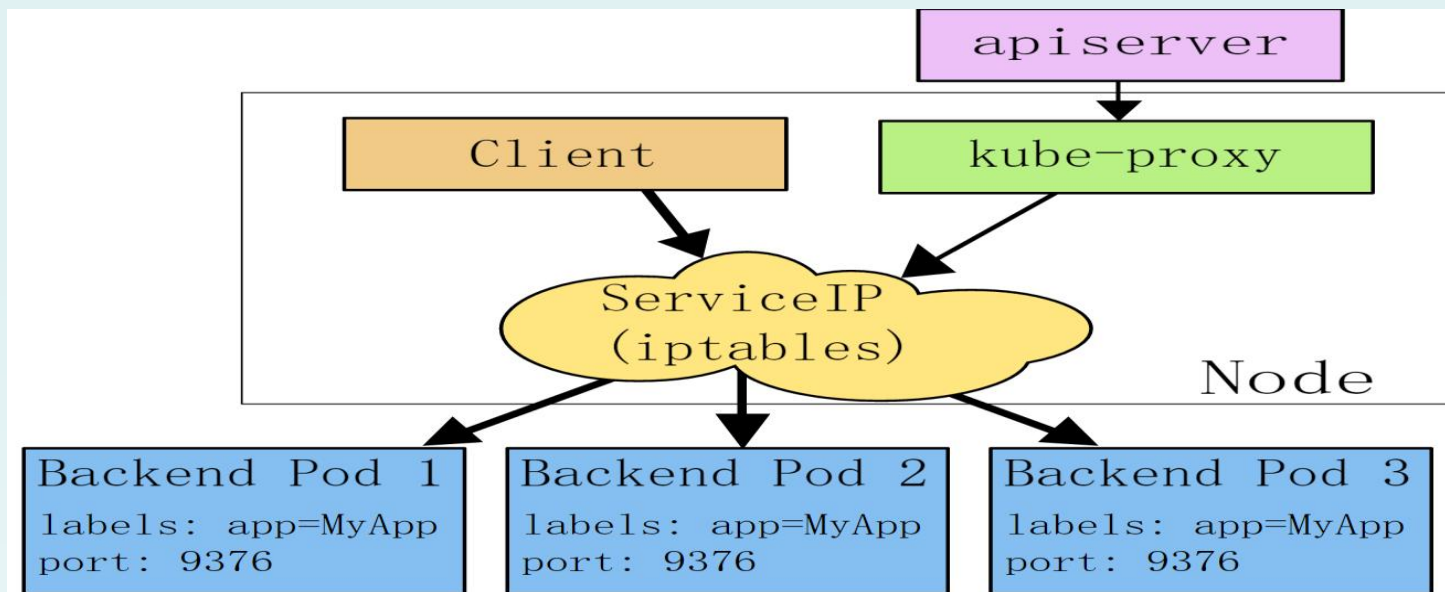
所有的容器都可以在不用NAT的方式下同别的容器通讯；

所有节点都可在不用NAT的方式下同所有容器通讯；

容器的地址和别人看到的地址是同一个地址。

k8s负载均衡-service

pod生命周期短，状态不稳定，pod异常后新生成的pod ip会发生变化，之前pod的访问方式均不可达。通过service对pod做代理，service有固定的ip和port，ip:port组合自动关联后端pod，即使pod发生改变，kubernetes内部更新这组关联关系，使得service能够匹配到新的pod。



kube-dns

- kube-dns组件会监听etc中service的变化，及时更新自己的dns规则，这样以来，我们就可以在pod中通过dns调用集群中的service了。

问题

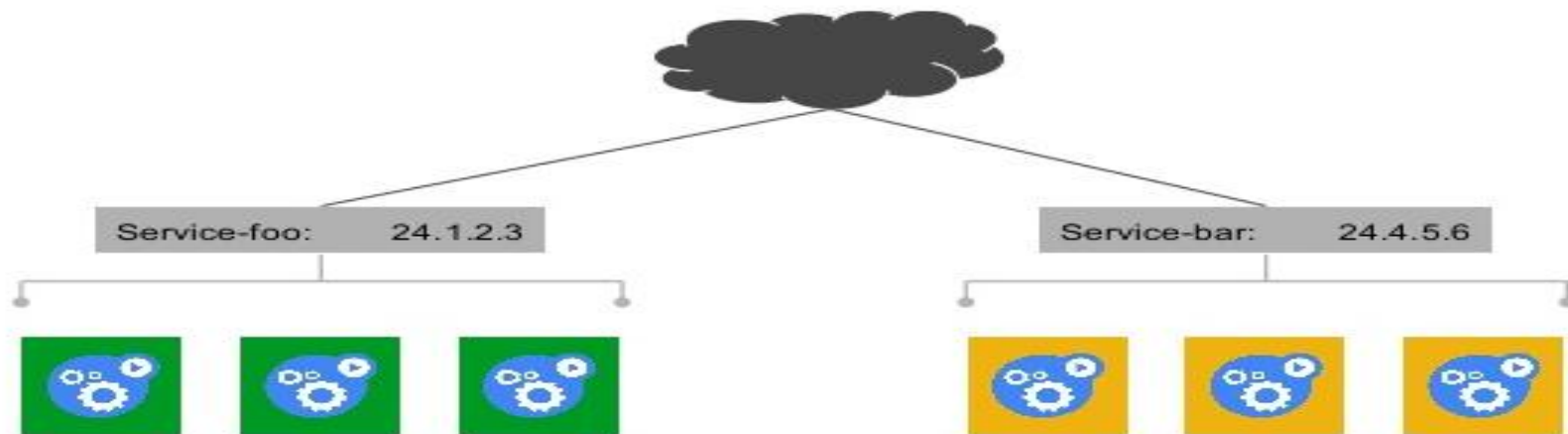
- 一只支持4层负载均衡，没有7层功能，没办法做一些高级的负载策略（cookie）
- 一 对外访问的时候，NodePort类型需要在外部搭建额外的负载均衡，而LoadBalancer类型要求kubernetes必须跑在支持的cloud provider上面
- 一 NodePort还会有端口冲突问题
- 一 主机需要开放外围一堆的端口进行服务调用，导致管理混乱

k8s负载均衡-ingress

Service和Pod的IP地址只能在集群内部网络中路由，所有到达“边界路由器”（Edge Router）的网络流量要么被丢弃，要么被转发到别处

Ingress: 是对外（公网）服务到集群内的Service之间规则的集合，允许进入集群的请求被转发至集群内

Ingress for HTTP Load Balancing [Beta]



- **Ingress**本身并不会自动创建负载均衡器，**cluster**中需要运行一个**ingress controller**来根据**Ingress**的定义来管理负载均衡器。目前社区提供了**nginx**和**gce**的参考实现。

问题

- 1、需要外置负载均衡支撑
- 2、配置部署复杂
- 3、负载均衡节点存在高可用问题
- 4、负载均衡性能问题
- 5、DNS配置问题
- 6、与外部应用的结合问题

完

