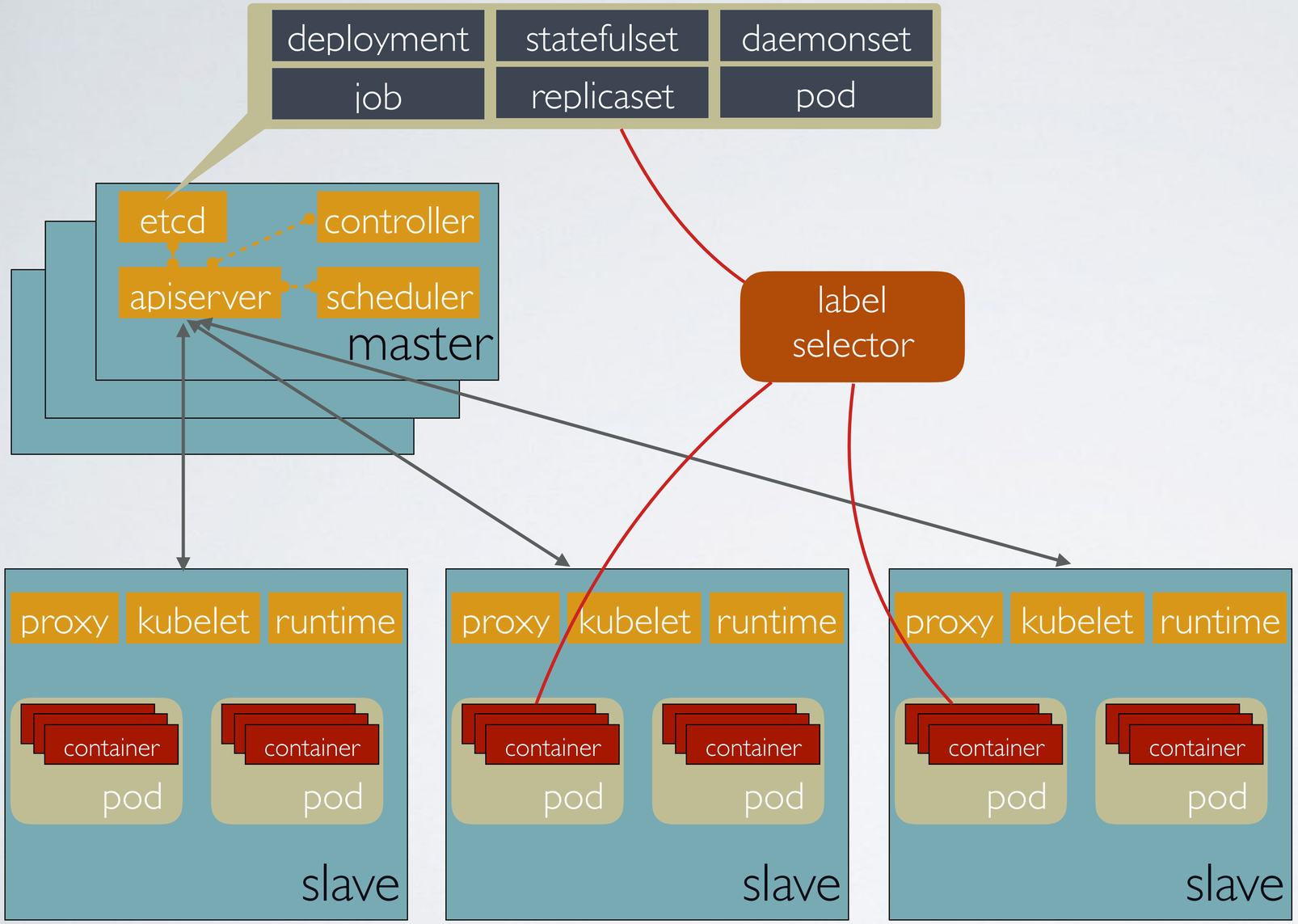
Kubernetes
应用流量管理介绍及实践

徙远
阿里巴巴系统软件事业部
Github:@huangyuqi

目录

Kubernetes应用模型和应用管理



- Kubernetes的伟大，一部分来自它的模型定义；起于borg，是个"富二代"；

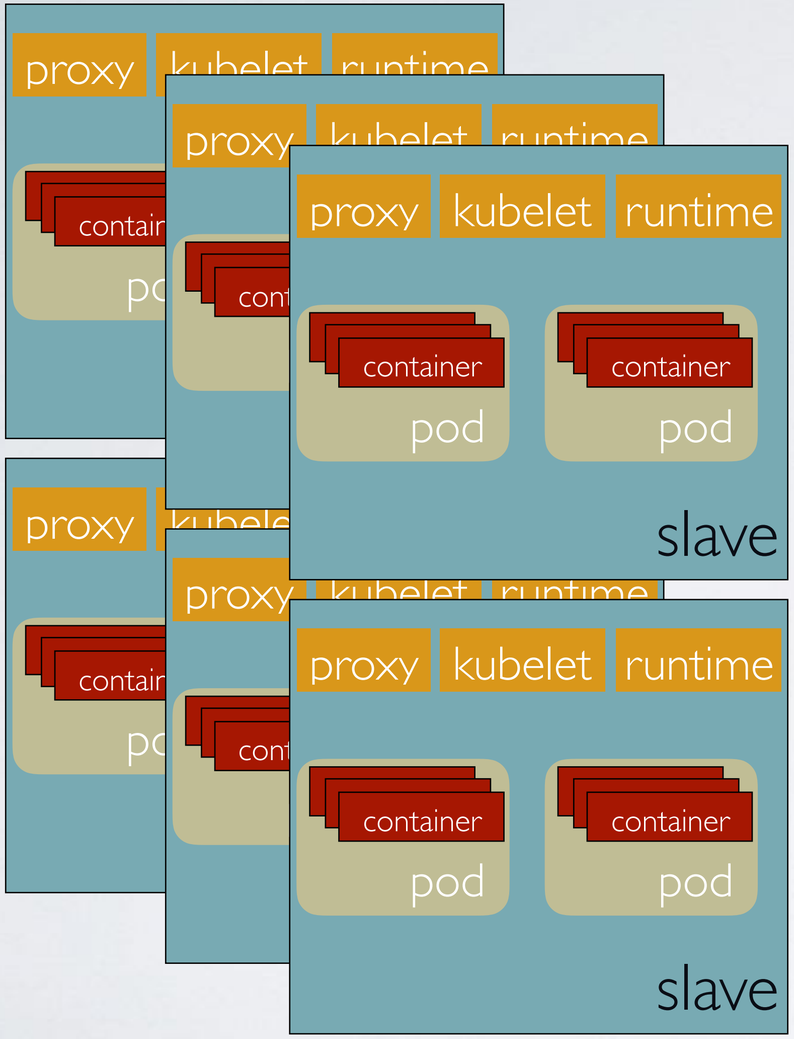
deployment	无状态应用
statefuleset	有状态应用
daemonset	守护进程类应用
job	任务类型应用

- Kubernetes在容器环境中是作为管理与部署引擎而存在的；
- 丰富的调度策略和健康检查机制
- 面向"终态"；基于模型的应用管理机制；

应用
生命周期管理方式的多样性和可靠性依赖于模型的抽象和各组件的协同，那么仅仅有应用就够吗？

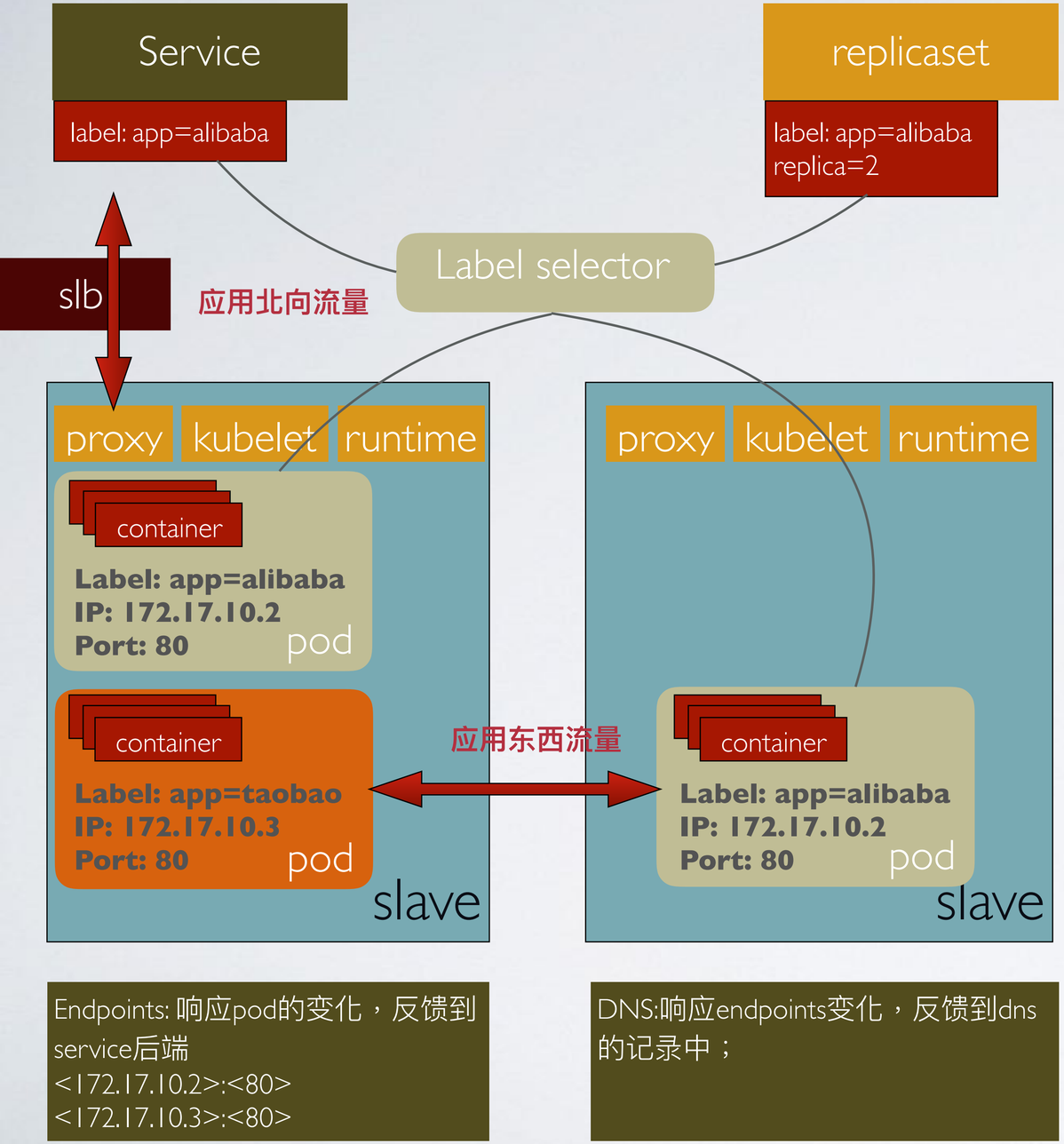
目录

Kubernetes应用/容器访问



- 如何访问归属各种应用模型的容器或POD？
- 多个后端副本实例，如何做到负载均衡，还要有四、七层能力？
- 容器迁移，IP发生变化了如何访问？
- 流量转发过程中的健康检查怎么做？
- 能不能通过域名访问？
- ……

Kubernetes应用访问-service

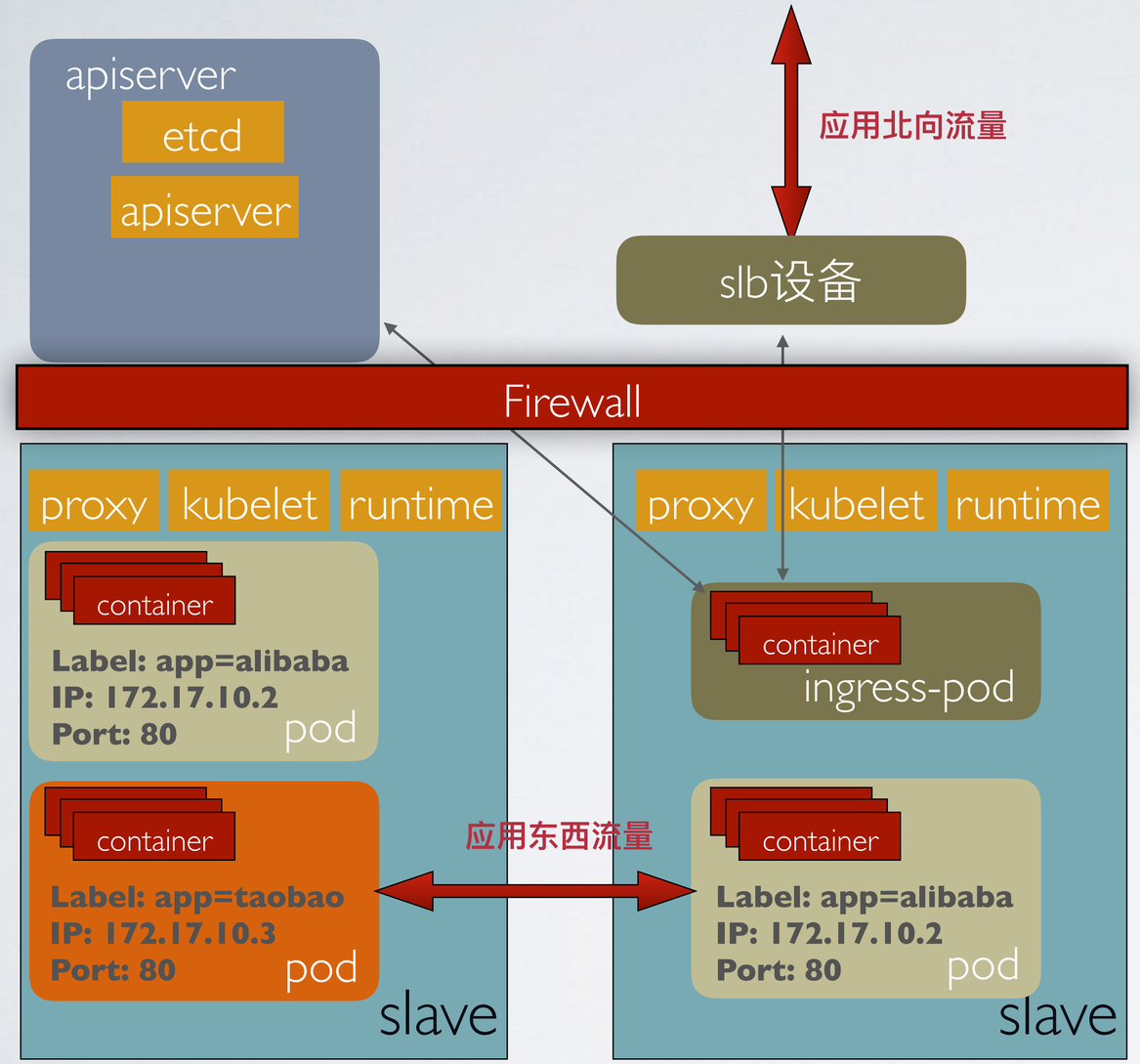


Service

- 提供POD访问的固定入口（IP+PORT形式）
  - 集群外可以通过nodeIP+nodeport（nodeport类型）
  - 集群外部还可以通过对接LB设备访问（loadbalancer类型）
  - 集群内部可以额外通过clusterIP+serviceport（clusterIP类型）
  - 集群内部还可以通过dns访问
- 提供多个POD副本间的LB；
- 屏蔽后端真实pod的变化，提供稳定的服务；
- Service是对一组提供相同功能的Pods的抽象，并为它们提供一个统一的入口

需要对外的服务过多以后，*nodeport*的维护开销将会显著增多，怎么办？如果能用域名是不是就解决了？包括一些七层的负责*slb*规则配置怎么下发？

Kubernetes应用访问-ingress



Ingress

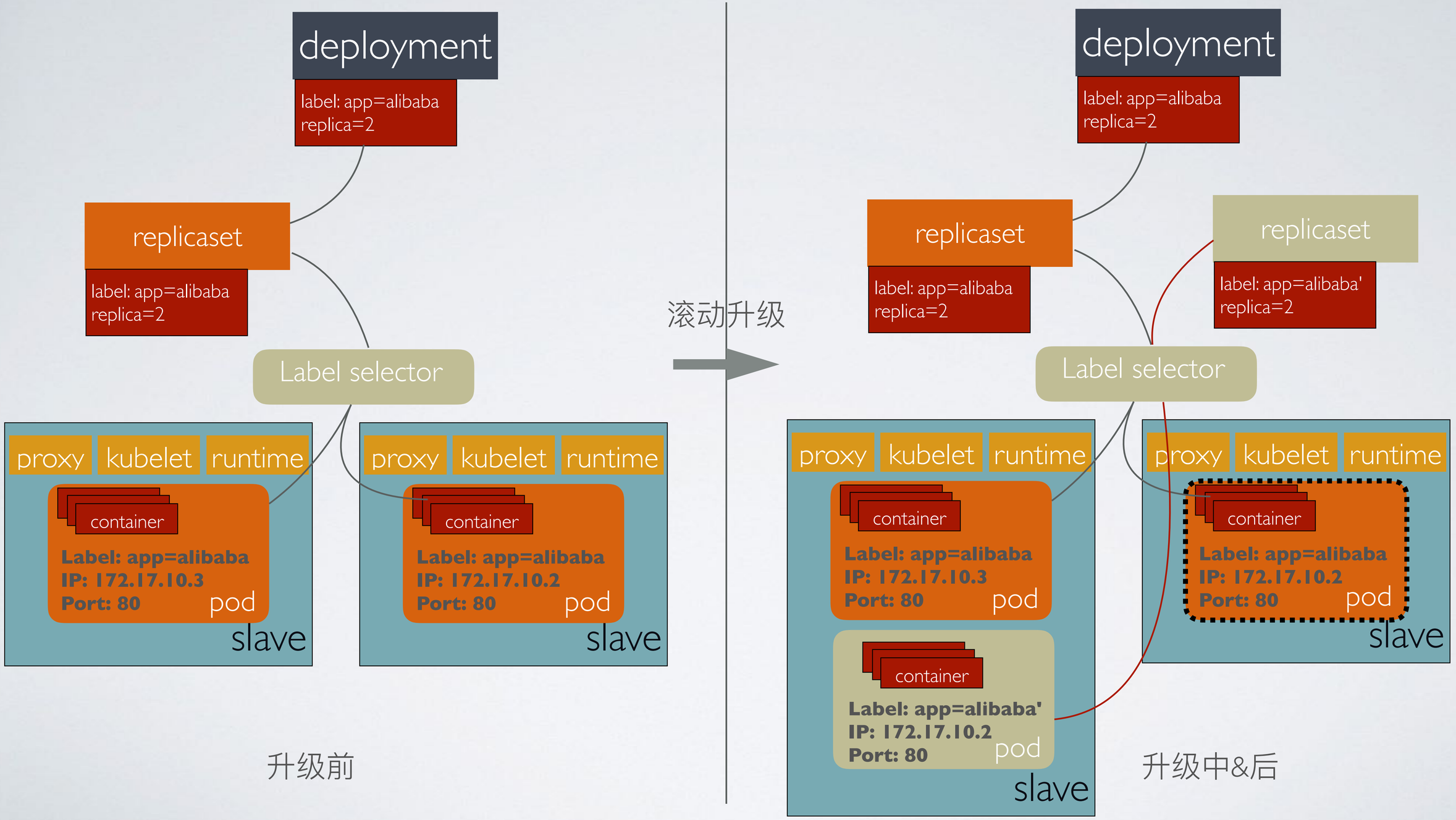
- Ingress是为进入集群的请求提供路由规则的集合
  - Ingress可以给service提供集群外部访问的URL、负载均衡、SSL终止、HTTP路由等

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - http:
      paths:
      - path: /testpath
        backend:
          serviceName: test
          servicePort: 80
```

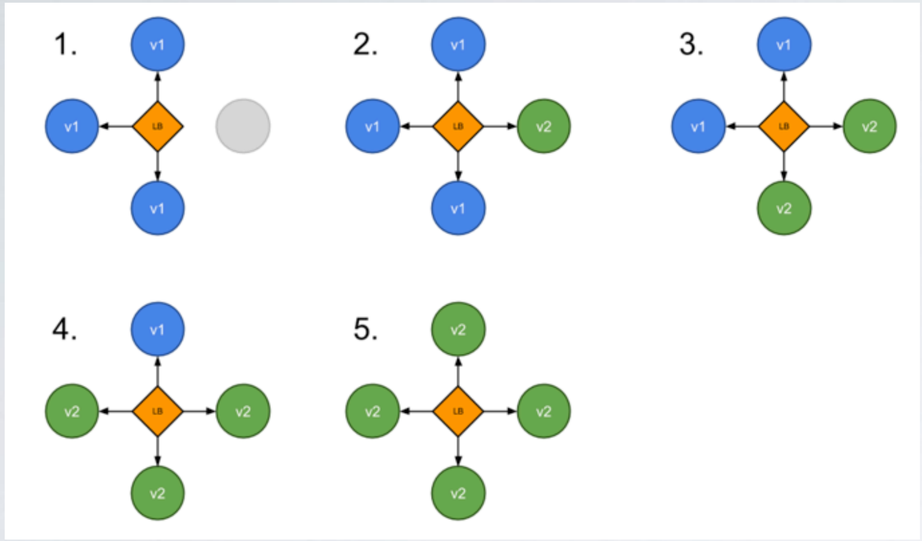
- 为了配置这些Ingress规则，集群管理员需要部署一个Ingress controller，它监听Ingress和service的变化，并根据规则配置负载均衡并提供访问入口

目录

Kubernetes原生无状态应用升级流程



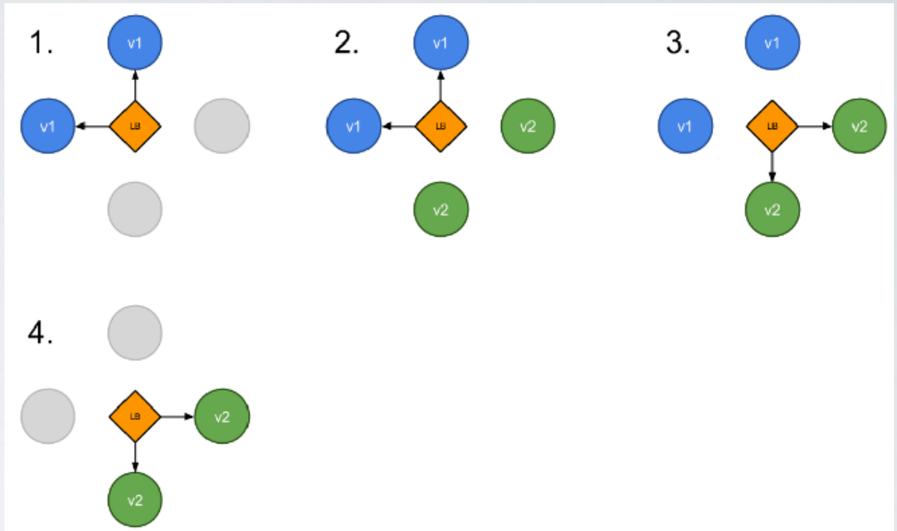
Kubernetes用户灰度发布需求



滚动：新老版本个数此消彼长，但是从流量层面不区分版本信息

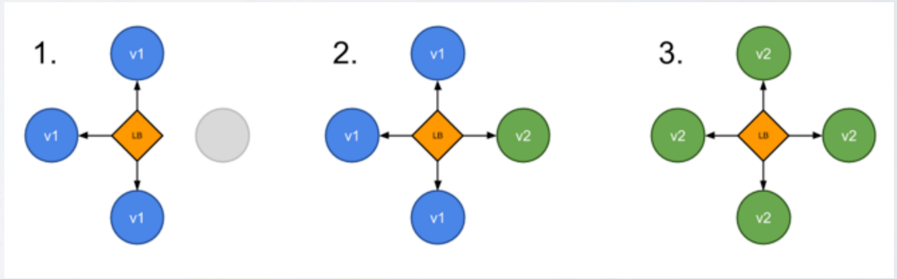
蓝绿：绿色版本测试完成后流量直接从蓝色版本切换至绿色版本；

＋



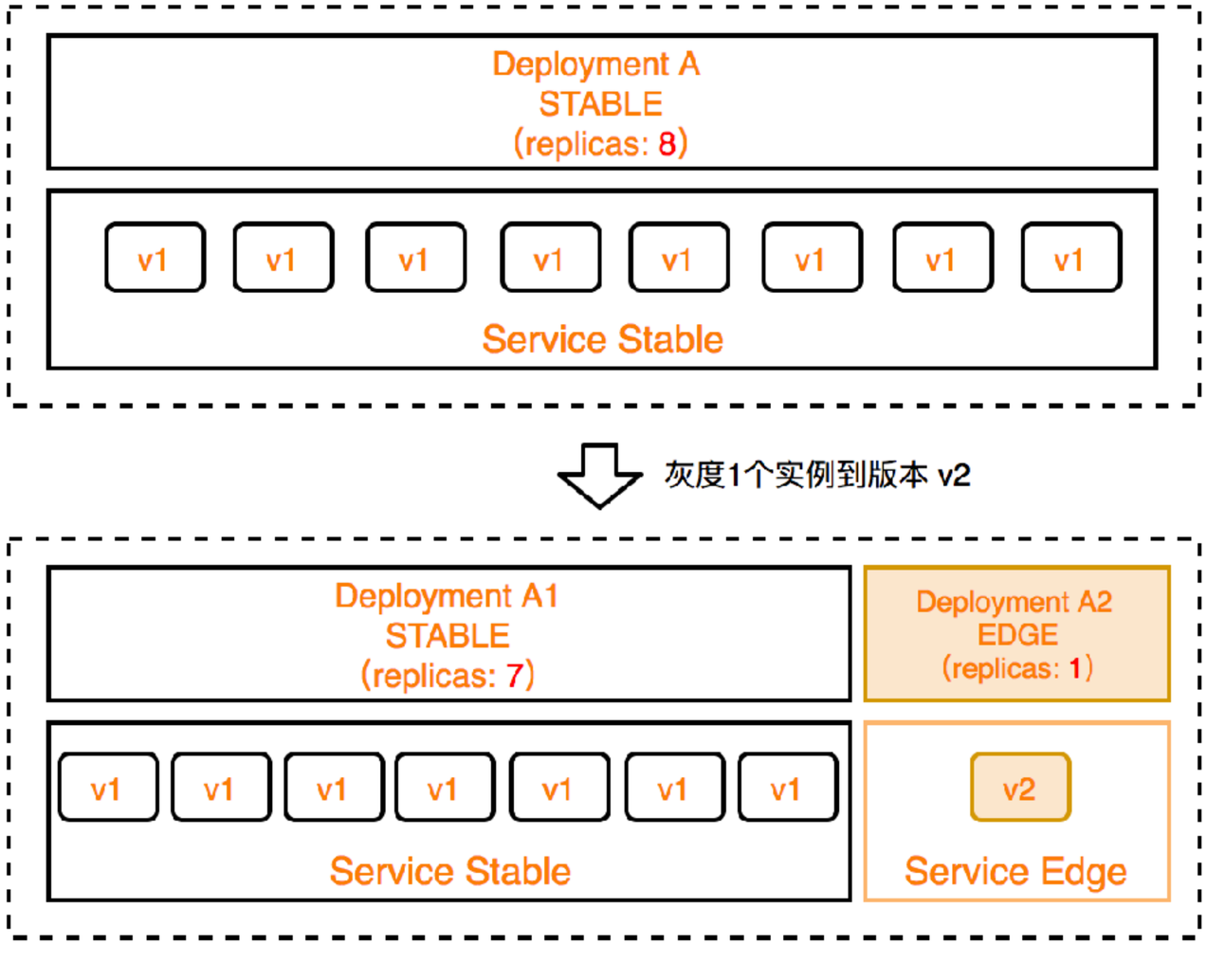
金丝雀：用户自测试； 当测试完成后将老版本升级至新版本

＝



灰度：灰度，是对应用新功能平滑过渡和验证的一种手段，灰度发布使得把应用的新功能版本部署到部分的节点中，灰度流量功能可以让特殊标记的流量引流到那些新功能版本的节点上，通过小流量方式验证功能。

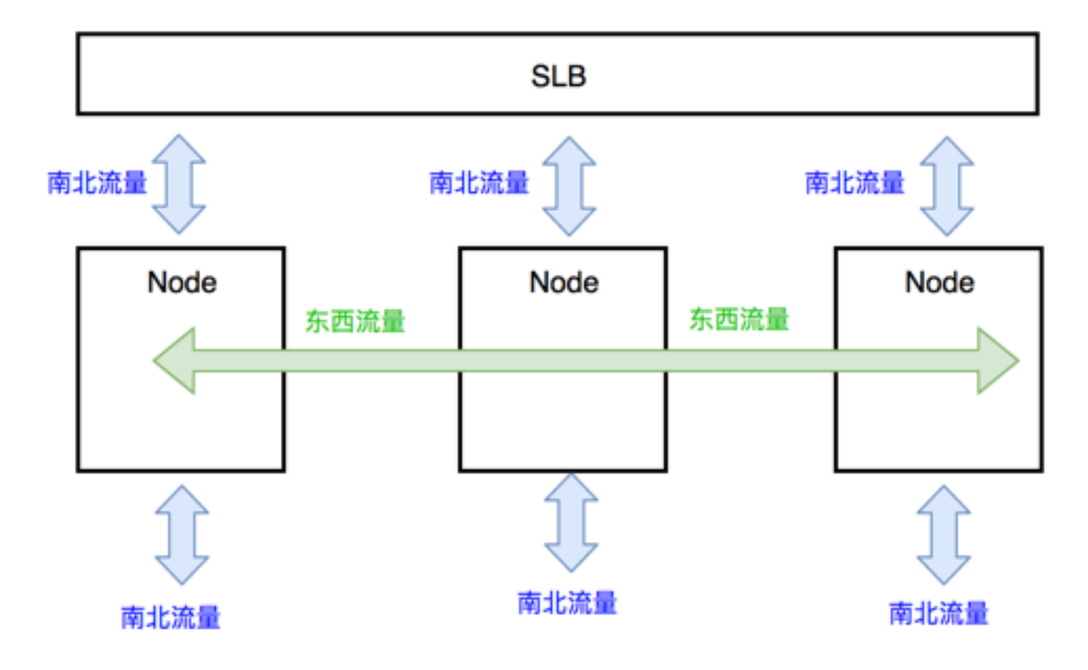
Kubernetes用户应用灰度



应用A灰度示意图

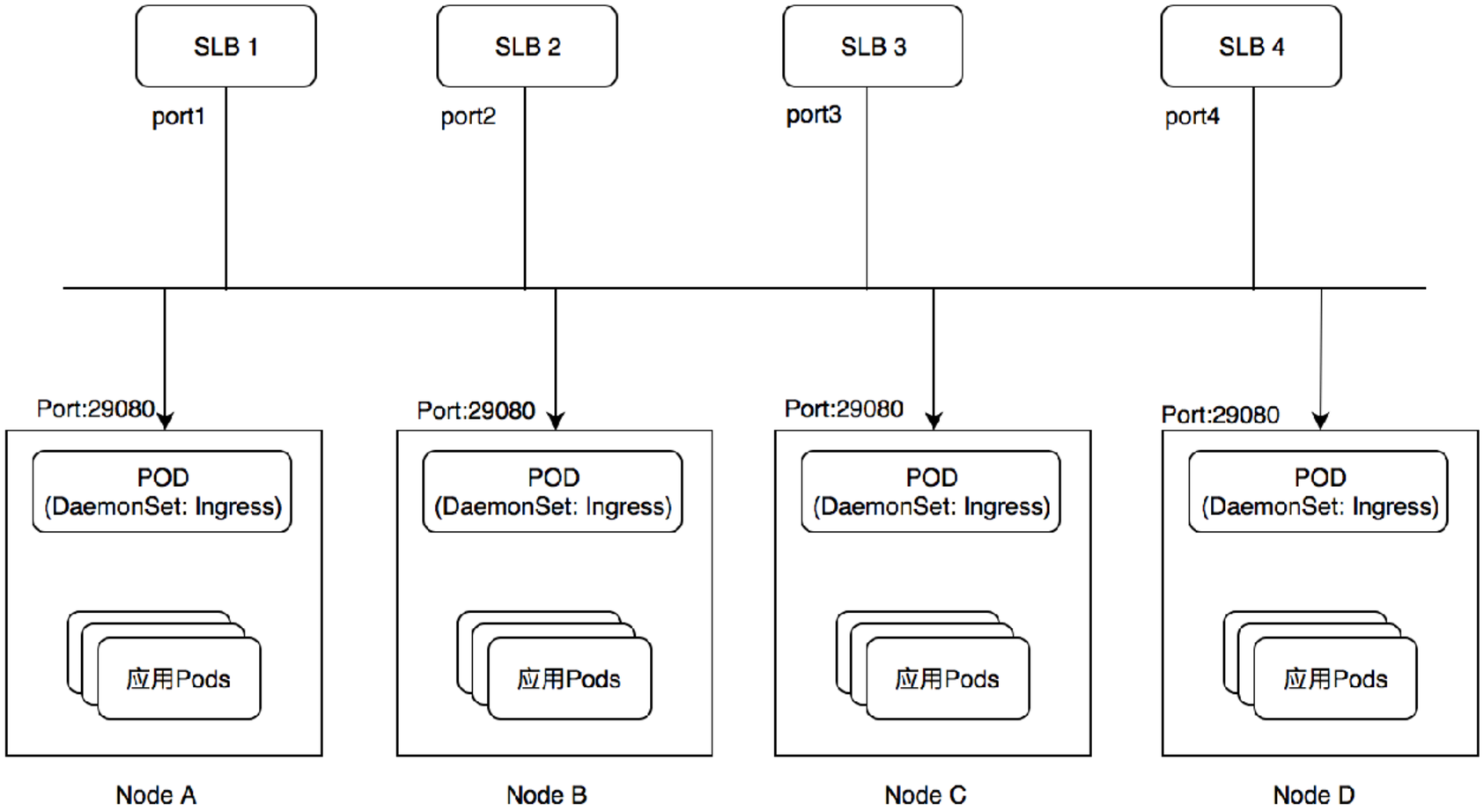
- 一个无状态的应用主要通过Deployment(RS->POD)完成部署，在我们目前的版本中，一个应用对应一个Deployment
- 通过额外的Deployment实现新功能版本的部署。一个应用在灰度的时候，同时具备两个Deployment，一个是stable版本的Deployment，一个是edge版本的Deployment。

Kubernetes用户灰度流量



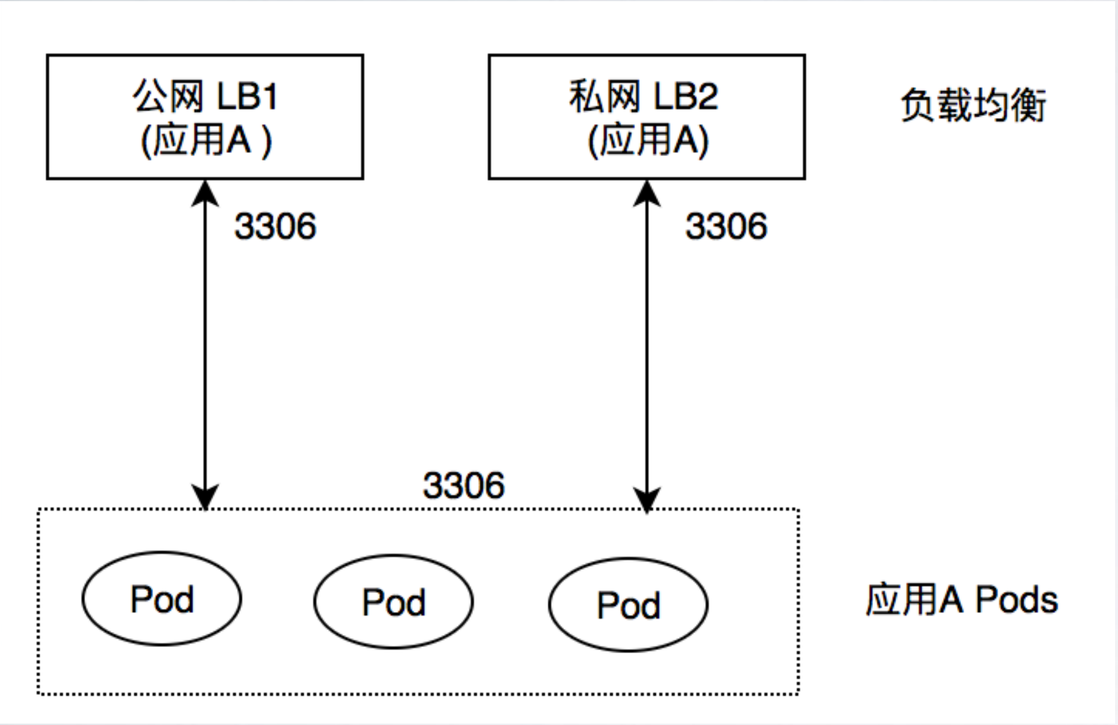
- 灰度流量主要是南北流量
- 南北流量通常通过service或者ingress实现访问；
- 灰度流量需要能够精细化控制，例如基于header的token，

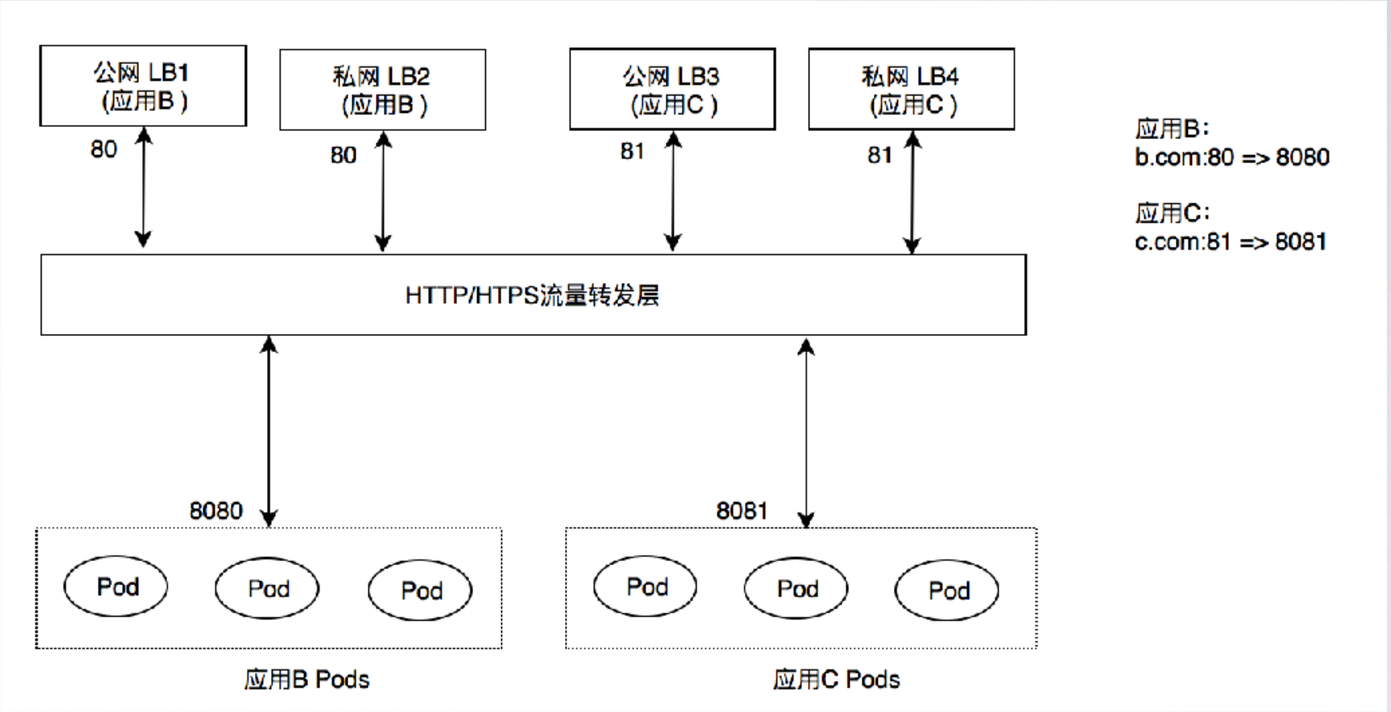
阿里云EDAS-kubernetes流量灰度实现



- 利用SLB做为外部LoadBalancer
- 七层流量转发通过了nginx-ingress-controller实现；
- nginx-ingress-controller 采用DaemonSet而不是Deployment部署
- 流量经转路径：
  - 请求流量由SLB进入。
  - SLB转发请求到所在集群的ECS的29080端口
  - nginx收到请求后，根据ServerName和Location，反向代理请求后应用Pod上。

阿里云EDAS-kubernetes流量支持场景

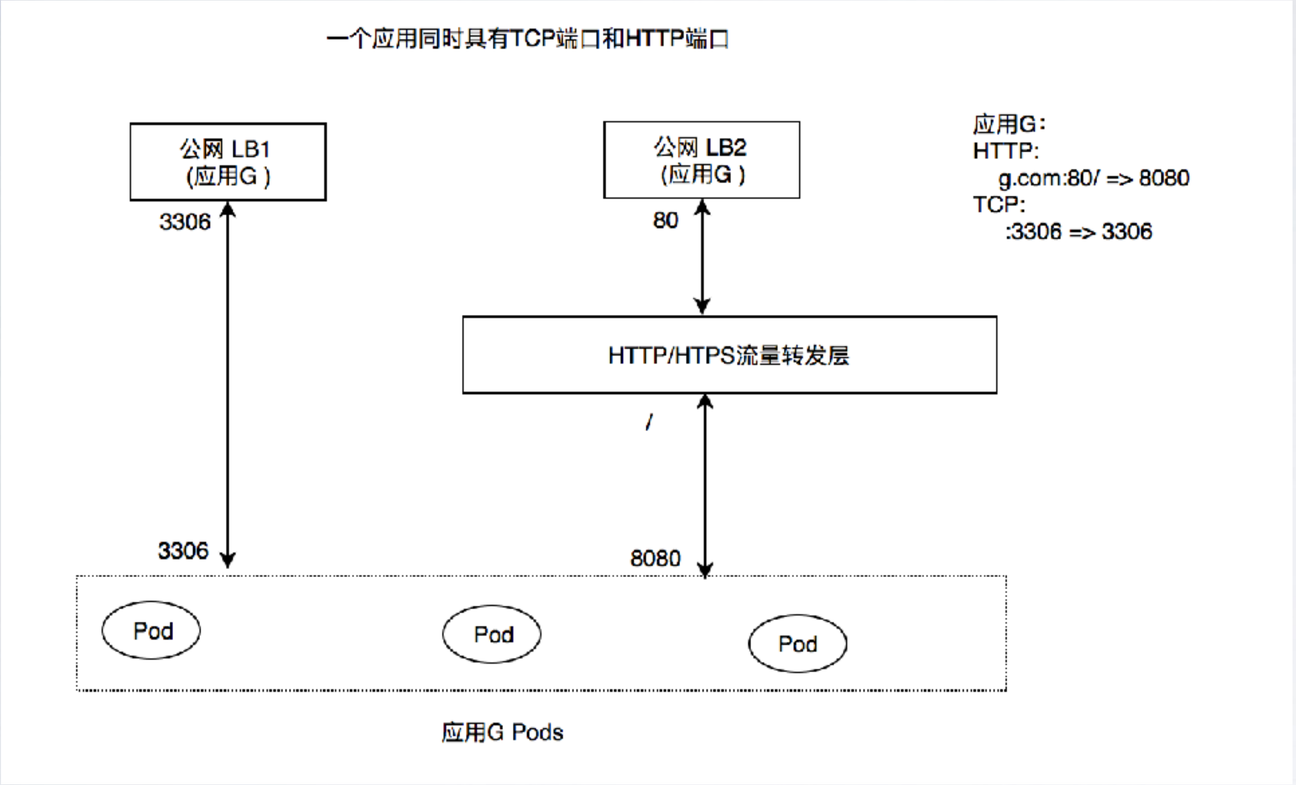




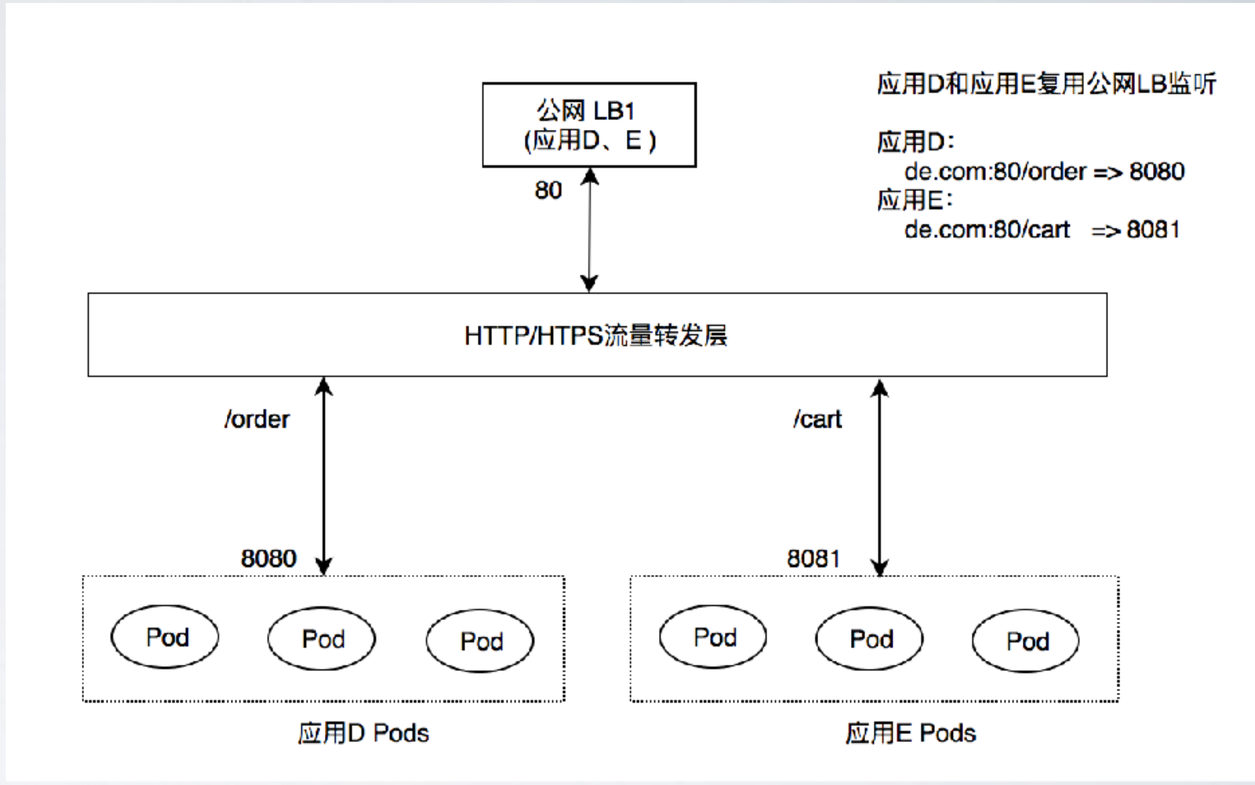
应用只有TCP端口，如mysql

应用只有一个HTTP端口

阿里云EDAS-kubernetes流量支持场景



应用同时具有TCP端口和HTTP端口



多个HTTP应用公用入口(LB和域名)
当流量从LB进入到流量转发层，会根据path分别分流到不同Pods组上

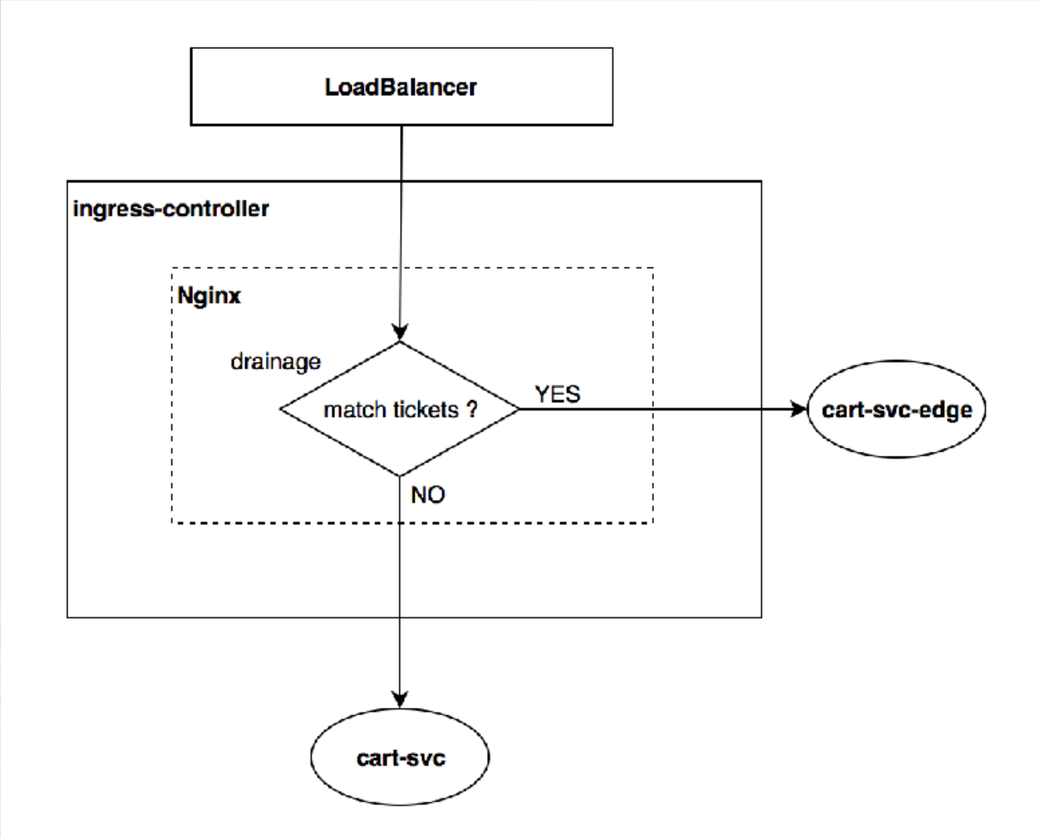
阿里云EDAS-kubernetes如何支持灰度流量

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
  name: taobao-ing
  namespace: default
spec:
  rules:
  - host: taobao.com
    http:
      paths:
      - backend:
          serviceName: cart-svc
          servicePort: 8080
        path: /cart
      - backend:
          serviceName: order-svc
          servicePort: 8080
        path: /order
      - backend:
          serviceName: www-svc
          servicePort: 8080
        path: /
```

```yaml
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/service-config: '{"hosts":{"6a69616e67626f31303030312e782e636f6dd41d8cd98f00b204e
  name: taobao-ing
  namespace: default
spec:
  rules:
  - host: 6a69616e67626f31303030312e782e636f6dd41d8cd98f00b204e9800998ecf8427e
    http:
      paths:
      - backend:
          serviceName: cart-svc
          servicePort: 8080
        path: /cart
      - backend:
          serviceName: order-svc
          servicePort: 8080
        path: /order
      - backend:
          serviceName: www-svc
          servicePort: 8080
        path: /
```

理想的效果就是，正常的流量依然会流入存量Pod上，特殊标记的流量会转发到新的Pod上，但很遗憾的是，Ingress 并不支持这种功能

阿里云EDAS-kubernetes如何支持灰度流量




```
{
  // hosts 用来实现多域名、域名别名功能，通过唯一key进行1:N映射。
  "hosts": {
    "6a69616e67626f31303030312e782e636f6dd41d8cd98f00b204e9800998ecf8427e": [
      "taobao.com"
    ]
  },
  // drainage 为灰度分流相关的配置
  "drainage": {
    // tickets用户灰度流量的标识，目前支持"header"，后续会支持"cookie"和“query”
    "tickets": {
      "header": {
        "foo": "bar"
      }
    },
    // 正式service和灰度service的映射关系
    "serviceMap": {
      "cart-svc": "cart-svc-edge"
    }
  }
}
```

- 请求的流转和上文中的“流量经转路径”相同，当判断tickets匹配的时候，请求会被转发到灰度的POD上，否则转发到存量POD上

阿里云EDAS-kubernetes如何支持灰度流量

阿里云EDAS-kubernetes如何支持灰度流量

阿里云EDAS-kubernetes如何支持灰度流量