

阿里云 × CLOUD NATIVE
COMPUTING FOUNDATION

云原生技术公开课

第 05 讲

应用编排与管理：核心原理

张振 阿里巴巴技术专家

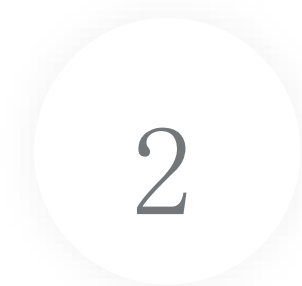


关注“阿里巴巴云原生”公众号
获取第一手技术资料





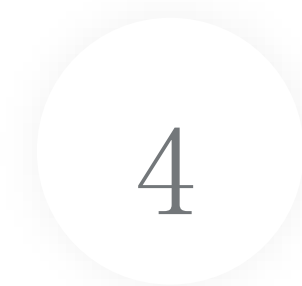
资源元信息



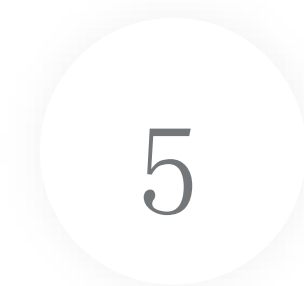
操作演示



控制器模式



控制器模式总结



课后思考和实践

Kubernetes 资源对象

- Spec: 期望的状态
- Status: 观测到的状态
- Metadata
 - Labels 
 - Annotations 
 - OwnerReference 

Labels

- 标识型的Key: Value元数据
- 作用
 - 用于筛选资源
 - 唯一的组合资源的方法
- 可以使用selector来查询
 - 类似于 SQL ‘select * where ...’

例子:

environment: production

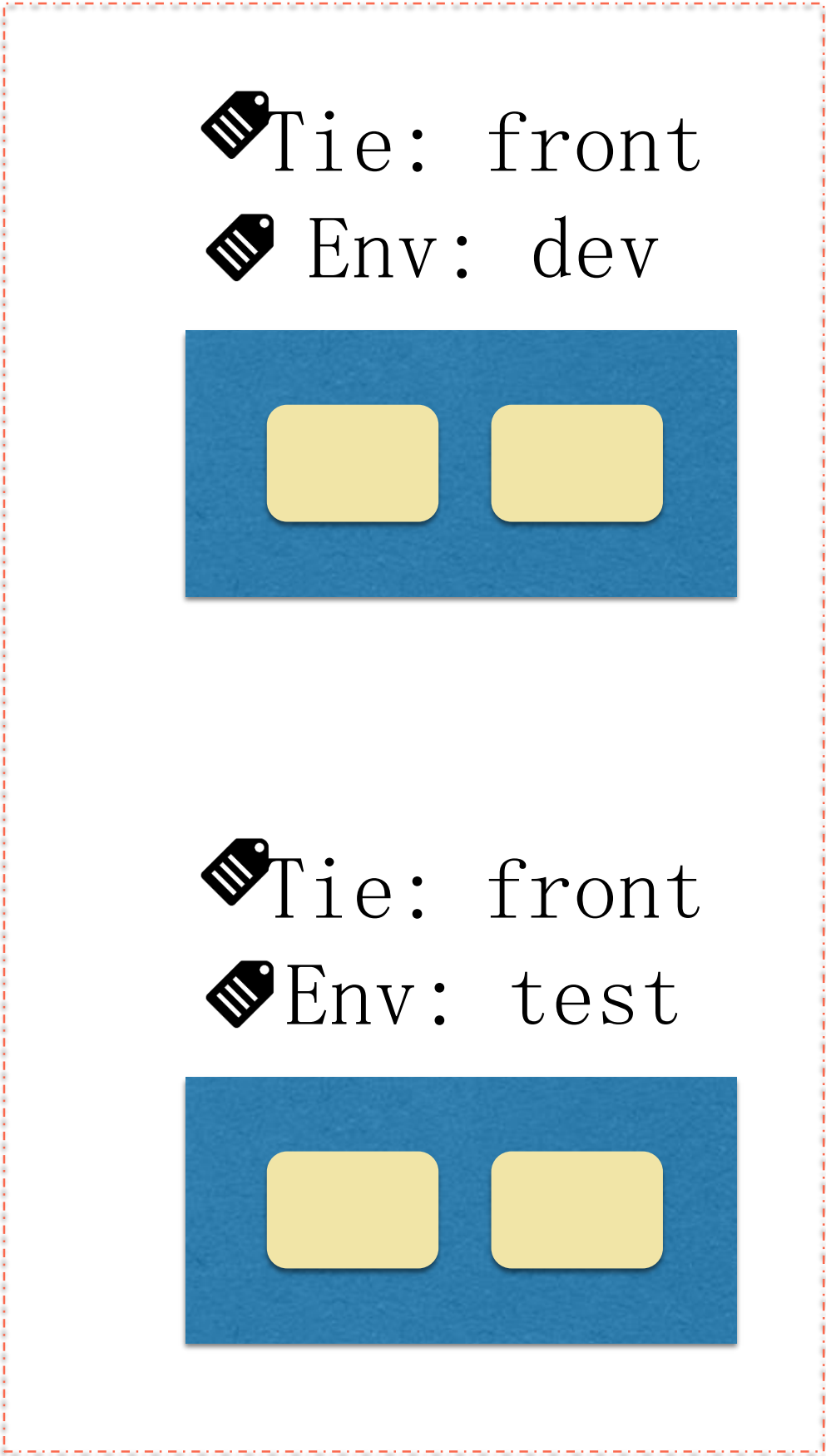
release: stable

app.kubernetes.io/version: 5.7.21

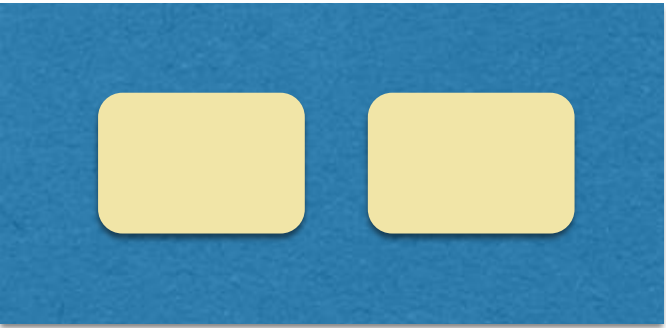
failure-domain.beta.kubernetes.io/region: cn-hangzhou

Selector

相等型Selector:
Tie=front



Tie: back
Env: prod

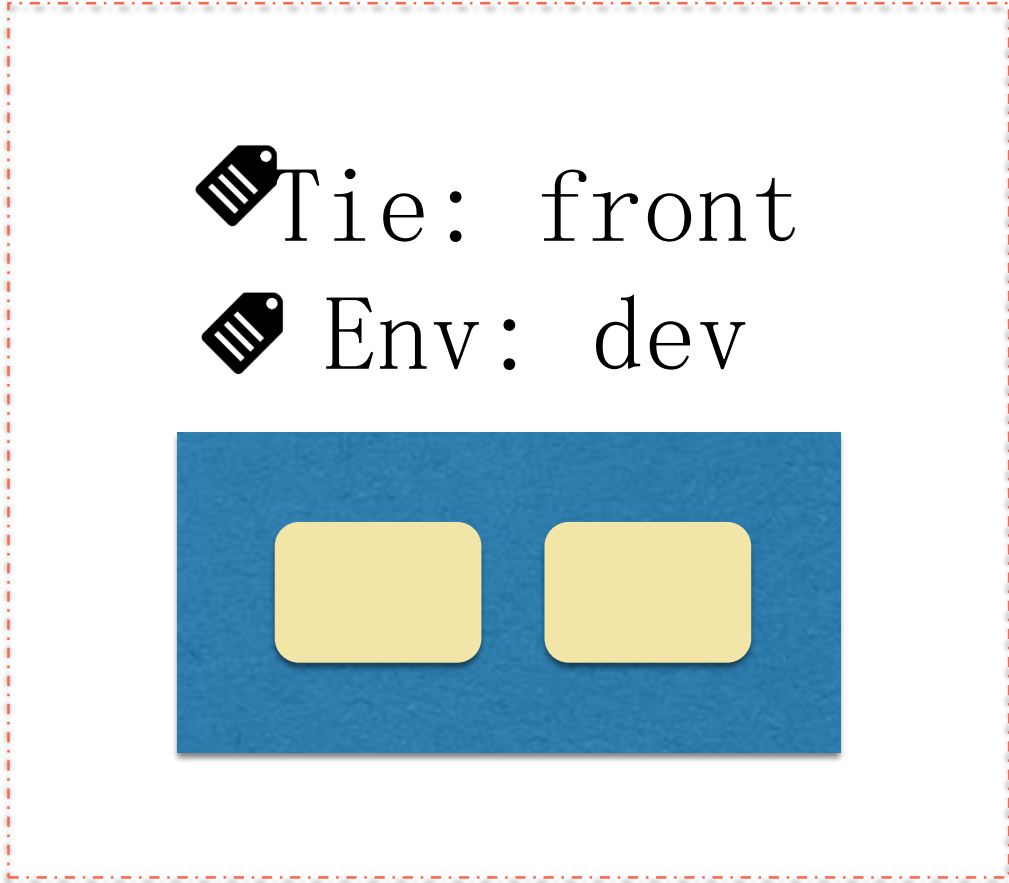


Tie: back
Env: gray



Selector

相等型Selector:
Tie=front, Env=dev



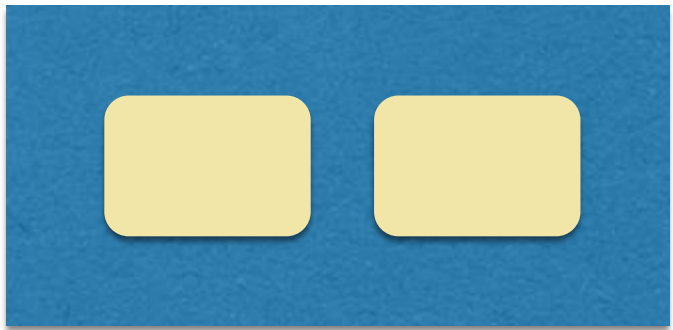
Tie: back
Env: prod



Tie: front
Env: test



Tie: back
Env: gray



Selector

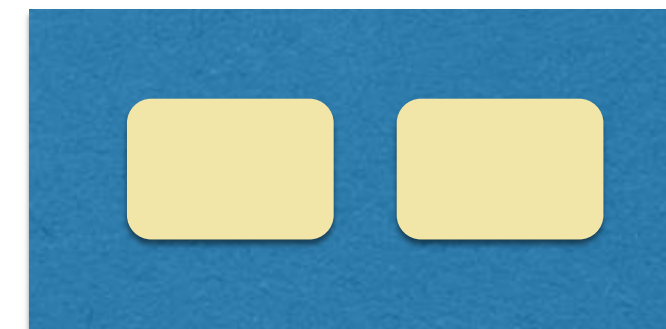
集合型Selector:

Env in (test, gray)

🏷️ Tie: front
🏷️ Env: dev



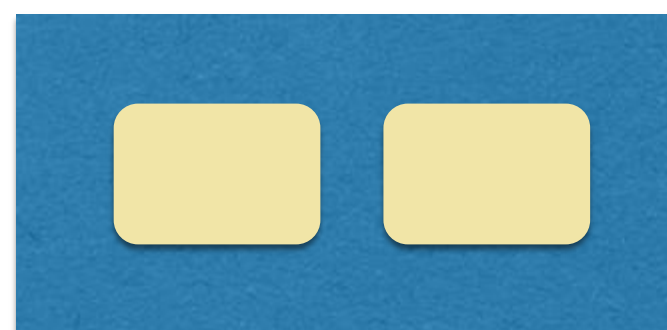
🏷️ Tie: back
🏷️ Env: prod



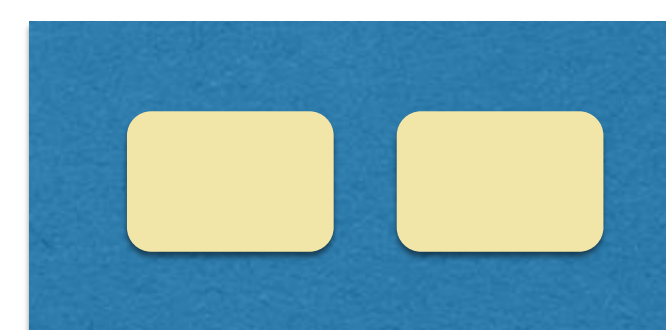
其他集合Selector例子:

- tie notin (front, back)
- release
- !release

🏷️ Tie: front
🏷️ Env: test



🏷️ Tie: back
🏷️ Env: gray



annotations

- Key: Value
- 作用:
 - 存储资源的非标识性信息
 - 扩展资源的spec/status
- 特点:
 - 一般比 label 更大
 - 可以包含特殊字符
 - 可以结构化也可以非结构化

例子:

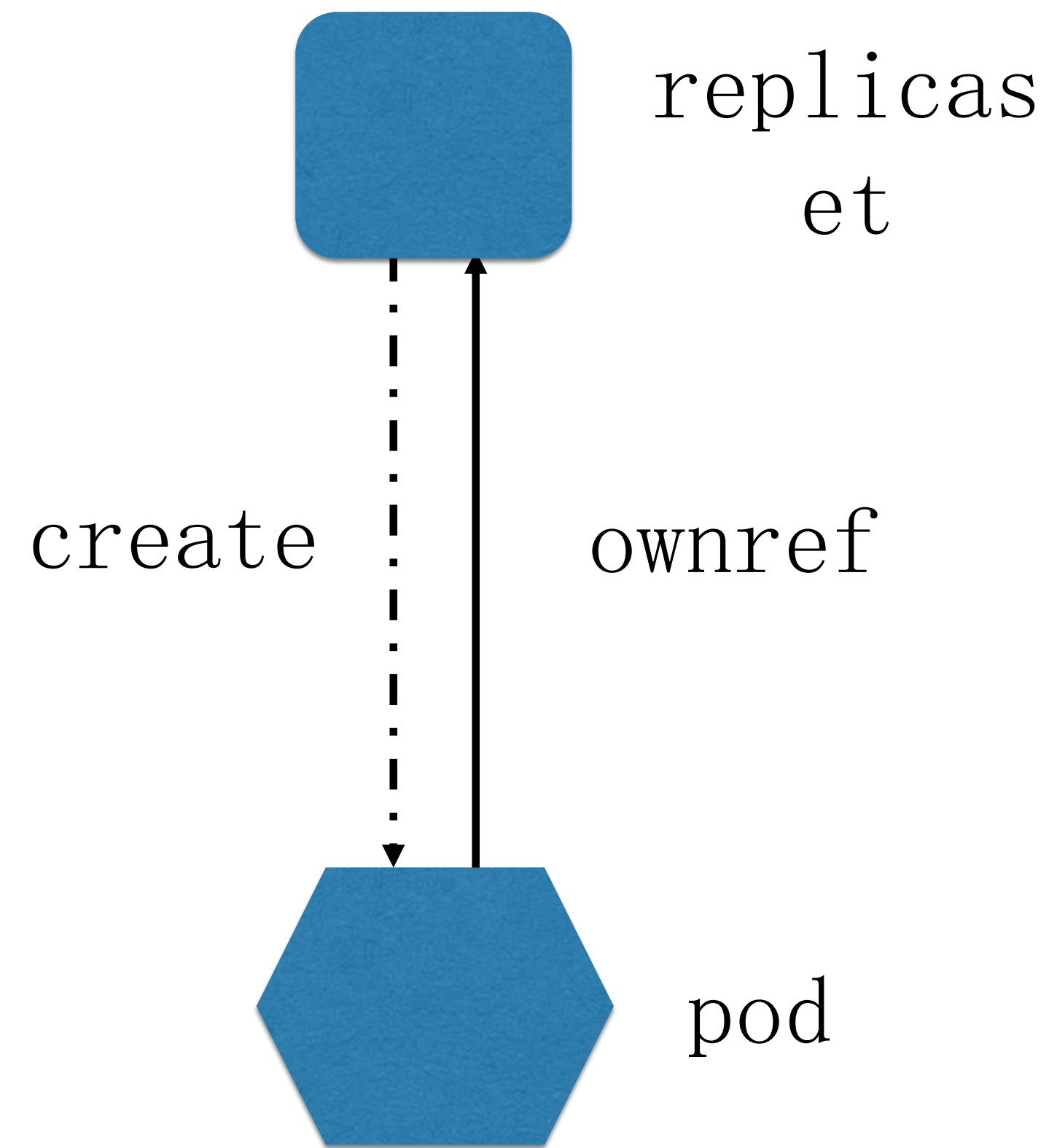
`service.beta.kubernetes.io/alibabacloud-loadbalancer-cert-id: your-cert-id`

`nginx.ingress.kubernetes.io/service-weight: "new-nginx: 20, old-nginx: 60"`

`kubectl.kubernetes.io/last-applied-configuration: /
{
 "apiVersion": "apps/v1",
 "kind": "Deployment",
 "metadata": {
 "annotations": {},
 "name": "nginx-deployment",
 "namespace": "default",
 "spec": {
 "selector": {
 "matchLabels": {
 "app": "nginx"
 }
 },
 "template": {
 "spec": {
 "containers": [{
 "image": "nginx:1.7.9",
 "name": "nginx"
 }]
 }
 }
 }
 }
}`

Ownreference

- “所有者”即集合类资源
 - **Pod** 的集合: **replicaset, statefulset**
- 集合类资源的控制器创建了归属资源
 - **Replicaset** 控制器创建 **pod**
- 作用
 - 方便反向查找创建资源的对象
 - 方便进行级联删除





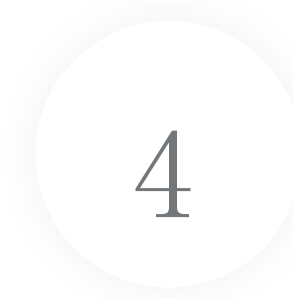
资源元信息



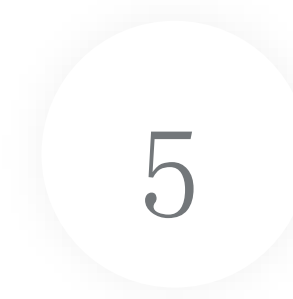
操作演示



控制器模式



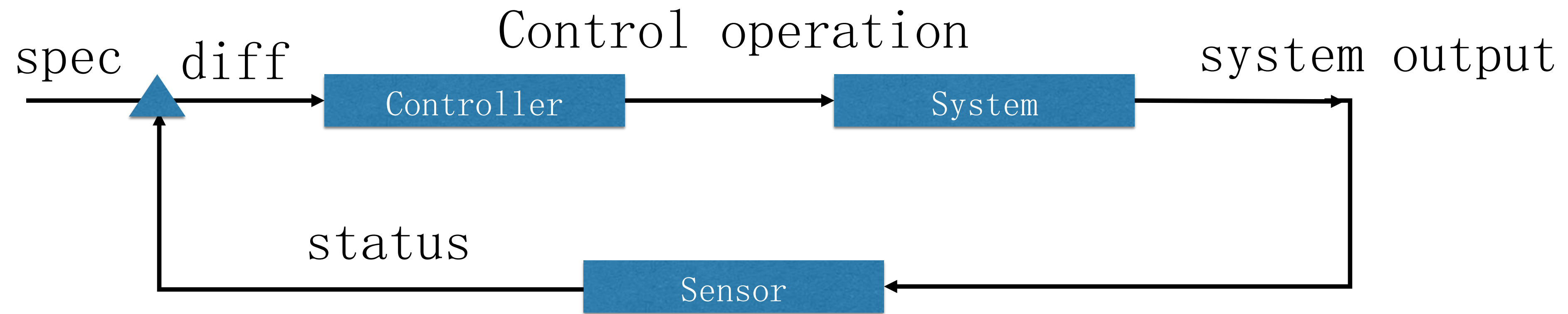
控制器模式总结



课后思考和实践

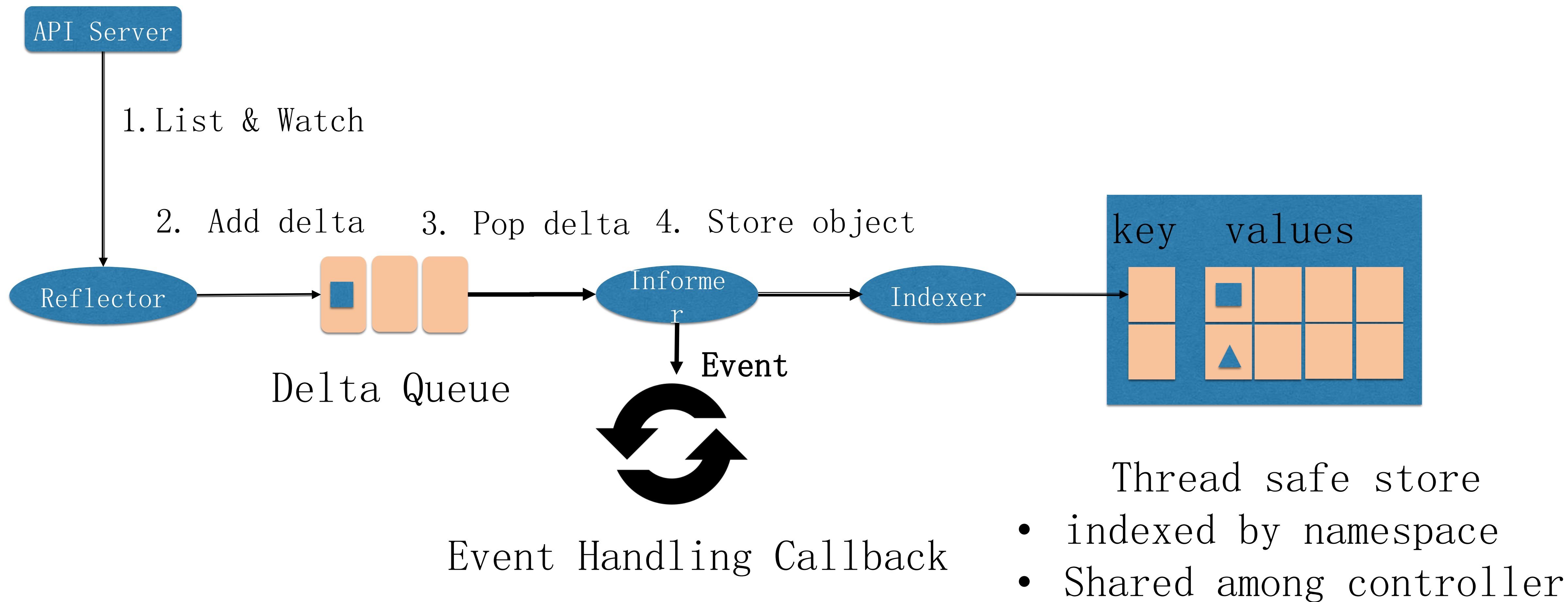


控制循环

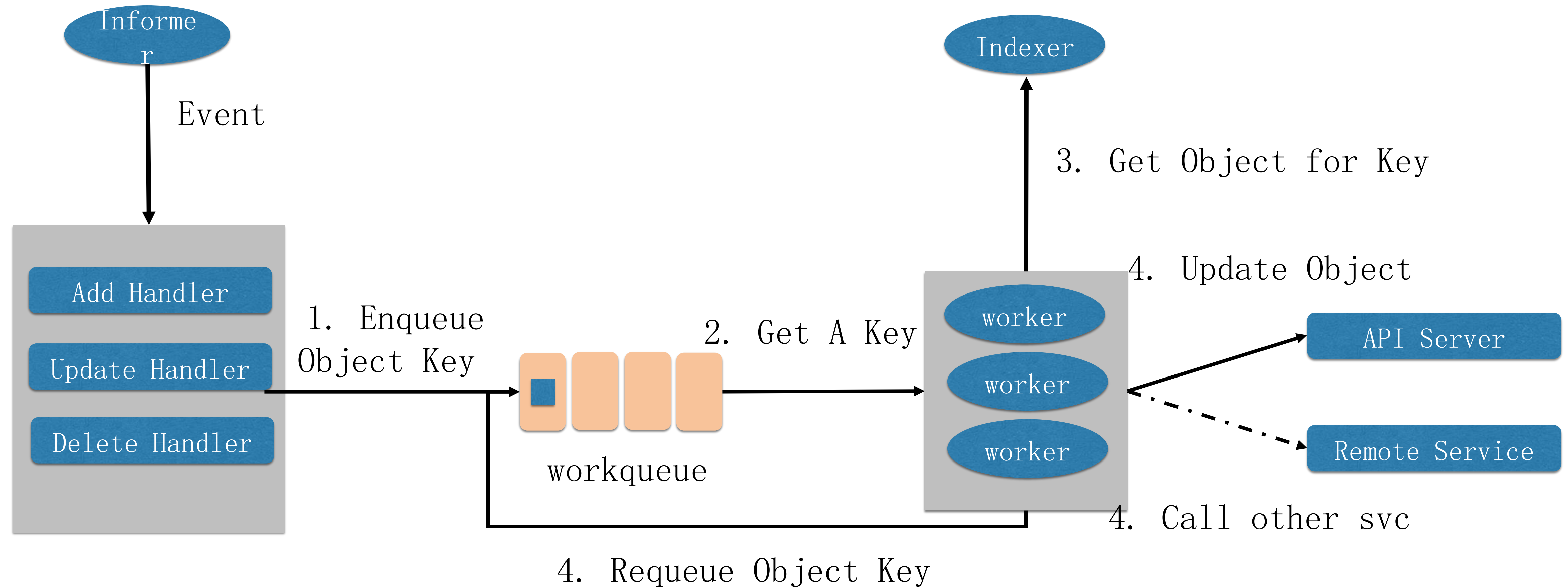


- 各组件独立自主地运行
- 不断使系统向终态趋近 **status -> spec**

sensor



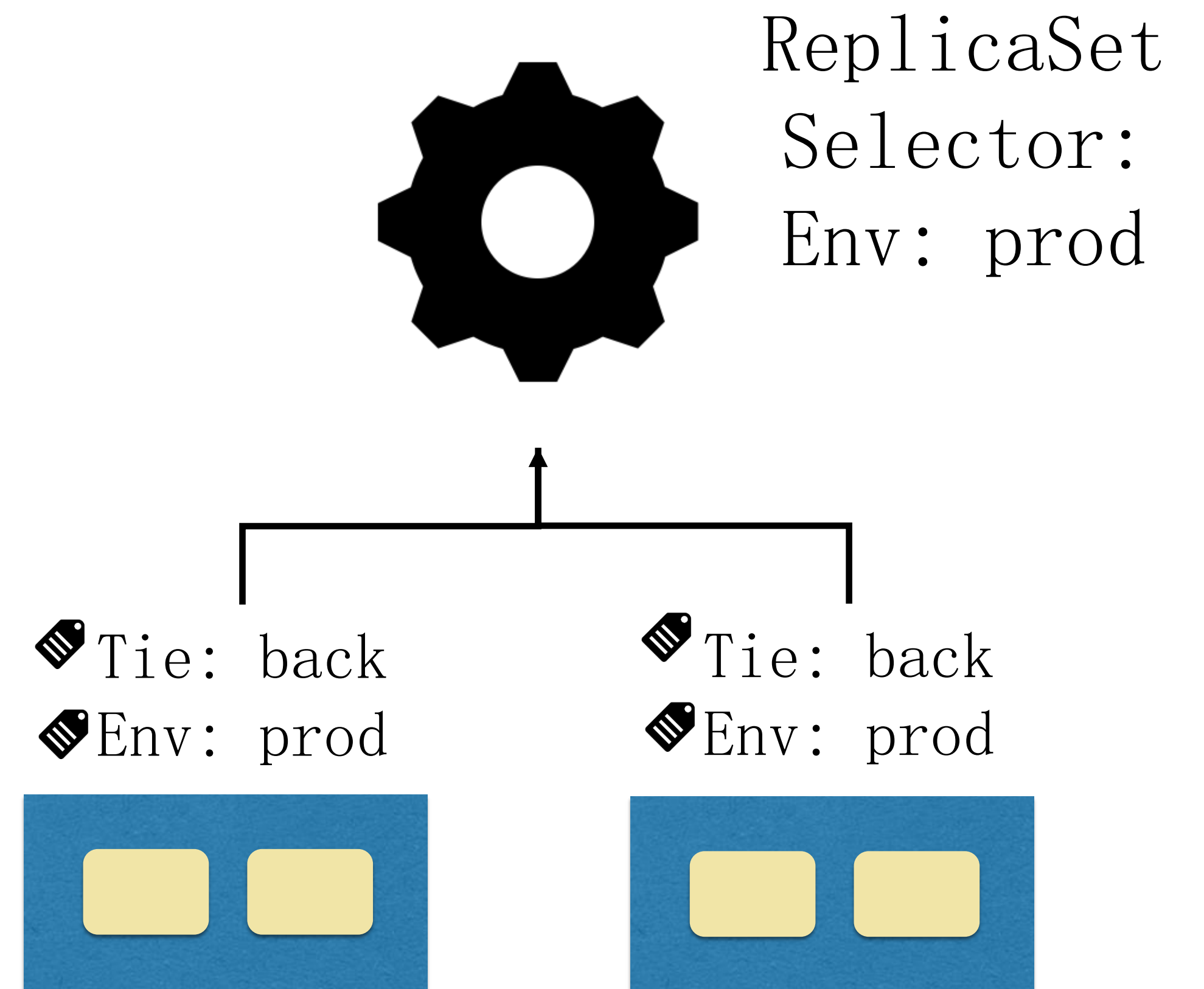
controller

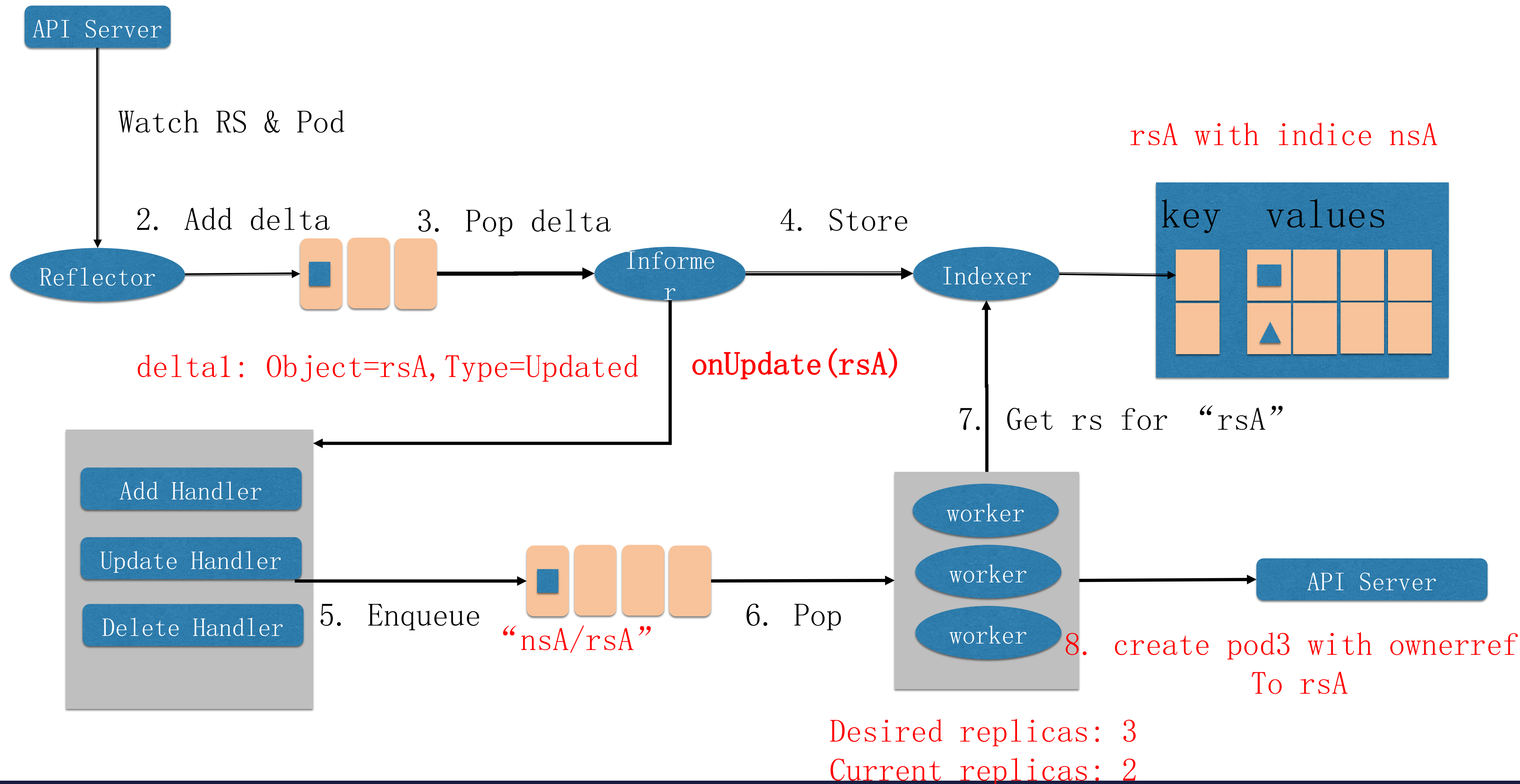


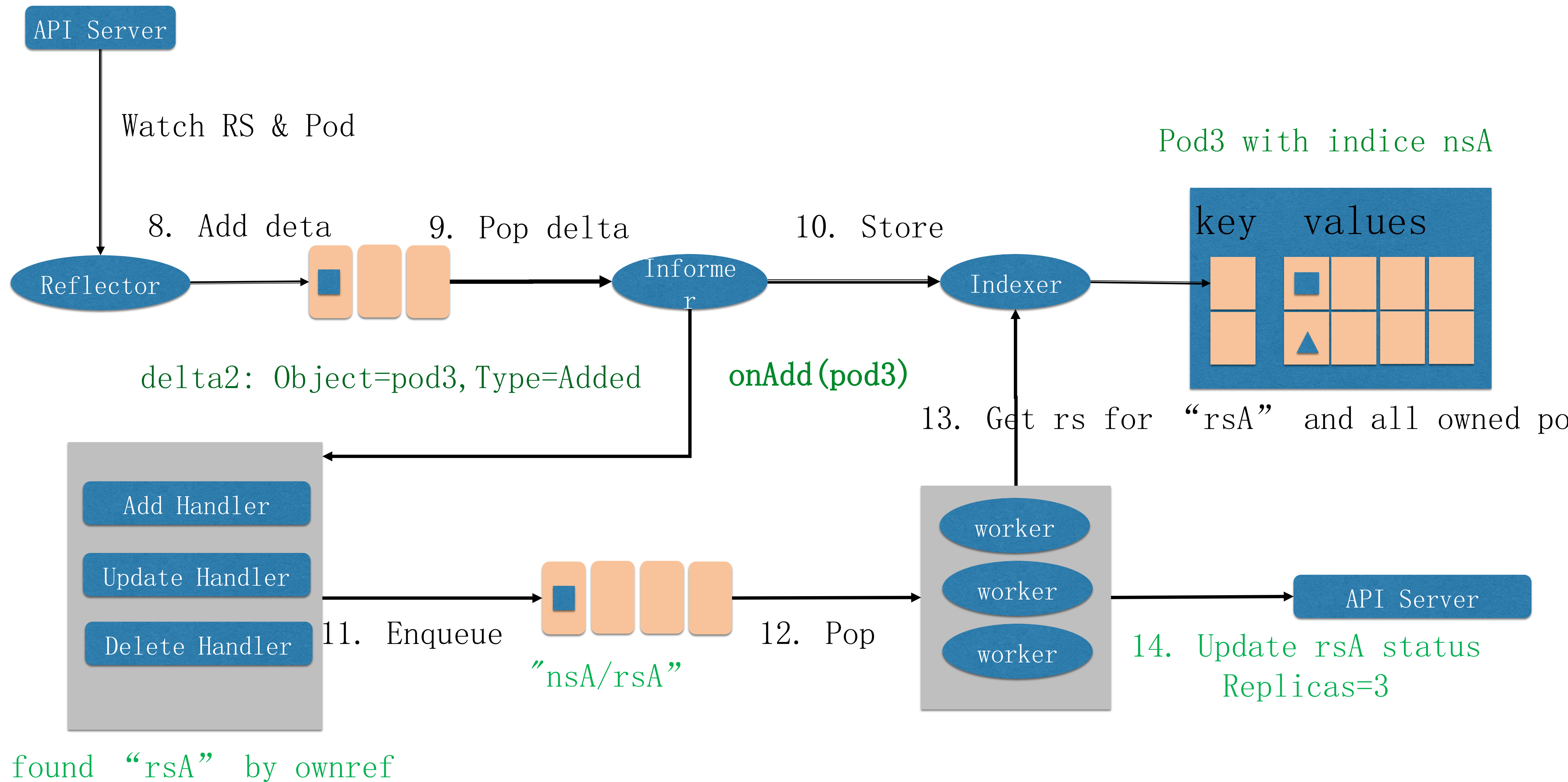
控制循环例子-扩容

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: rsA
  namespace: nsA
spec:
  replicas: 2
  selector:
    matchLabels:
      env: prod
  template:
    metadata:
      labels:
        env: prod
    spec:
      containers:
        - image: registry.cn-hangzhou.aliyuncs.com/acs/nginx
          name: nginx
status:
  replicas: 2
```

2 -> 3









两种 API 设计方法



命令式
(和孩子交流)

- 吃饭
- 刷牙
- 睡觉
- 唱一首歌
- 新扩一个 pod
- 删除一个 pod



声明式
(和员工交流)

- 市场占有率达到80%
- 稳定性达到99.99%
- 做一个身高体重正常的孩子
- 副本数保持在3个

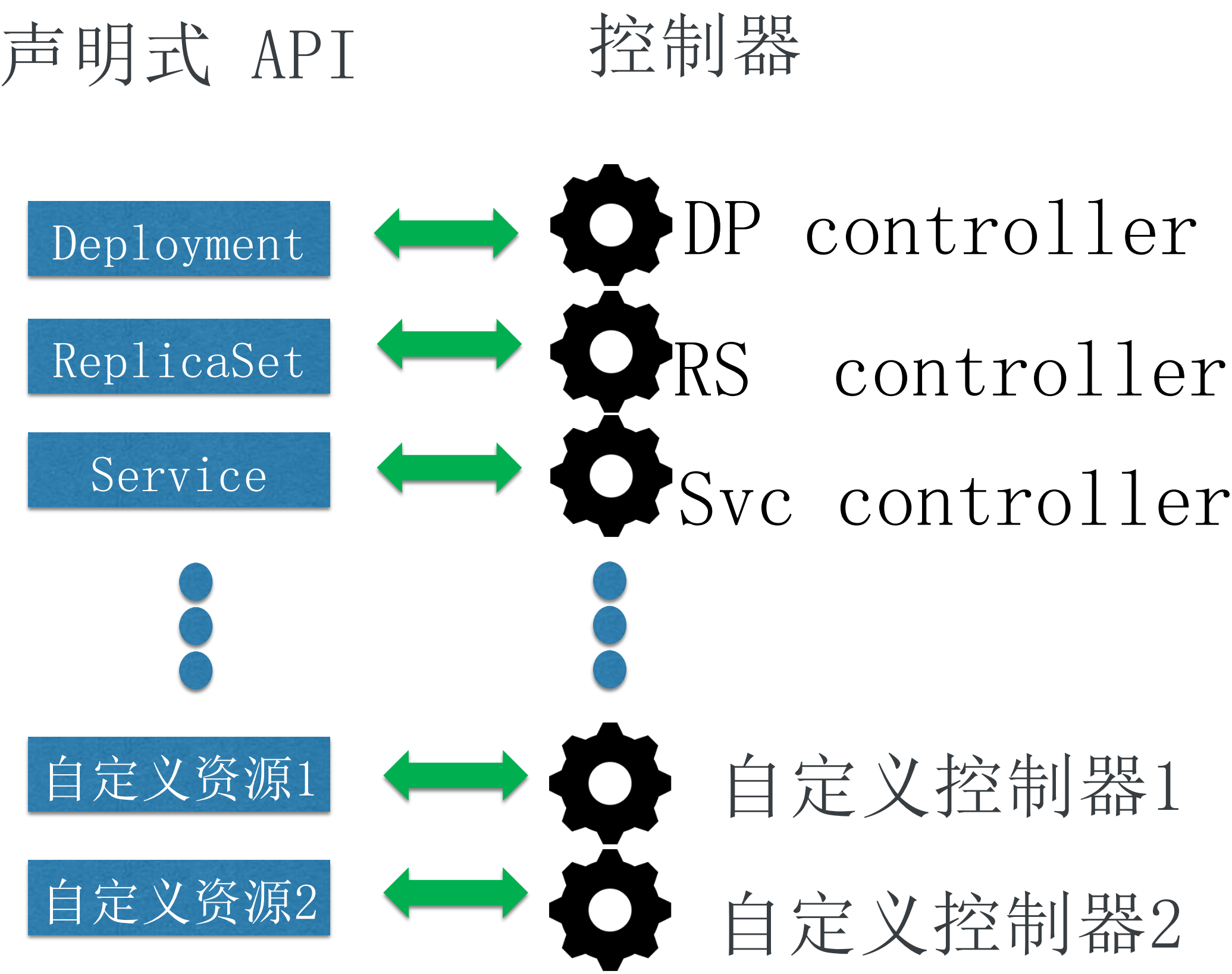
命令式

- 如果命令没有响应怎么办？
 - 反复重试
 - 需要记录当前的操作- 复杂
- 如果多重试了怎么办？
 - 巡检做修正- 额外工作、危险
- 如果多方并发访问怎么办？
 - 需要加锁- 复杂、低效

声明式

- 天然地记录了状态
- 幂等操作、可在任意时刻反复操做
- 正常操作即巡检
- 可合并多个变更

控制器模式总结



- 由声明式的 API驱动 - K8S资源对象
- 由控制器异步地控制系统向终态驱近
- 使系统的自动化和无人值守化成为可能
- 便于扩展-自定义资源和控制器
(特别的, operator)





关注“阿里巴巴云原生”公众号
获取第一手技术资料