HUAWEI

CloudNative Lives

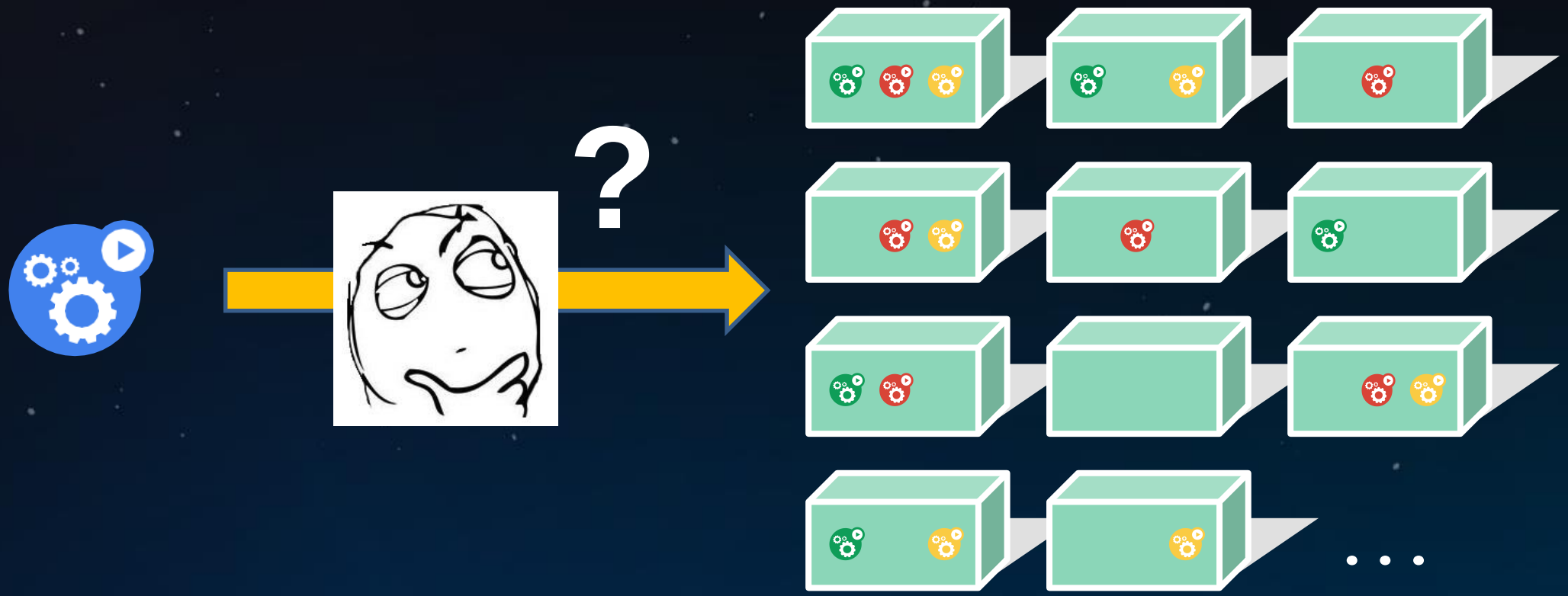
Kubernetes调度器原理剖析&实践

华为云容器团队核心架构师 & CNCF社区主要贡献者倾心打造

大 纲

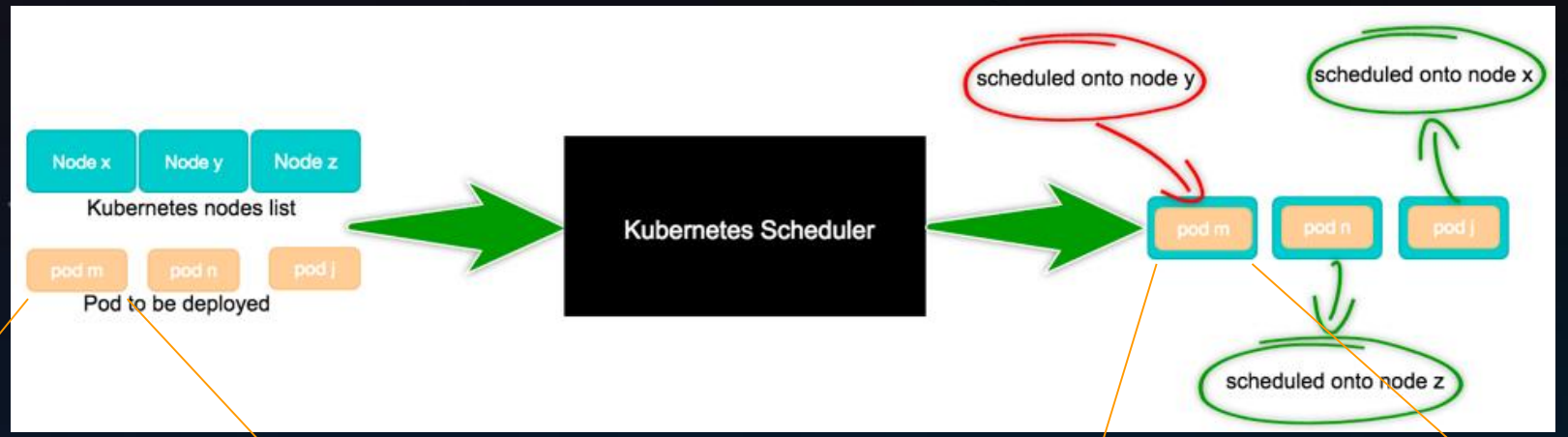
- **K8S调度机制介绍**
- K8S中的调度策略与算法
- K8S高级调度特性详解

Scheduler：为Pod找到一个合适的Node



茫茫人（Node）海，如何找到那个对的TA？

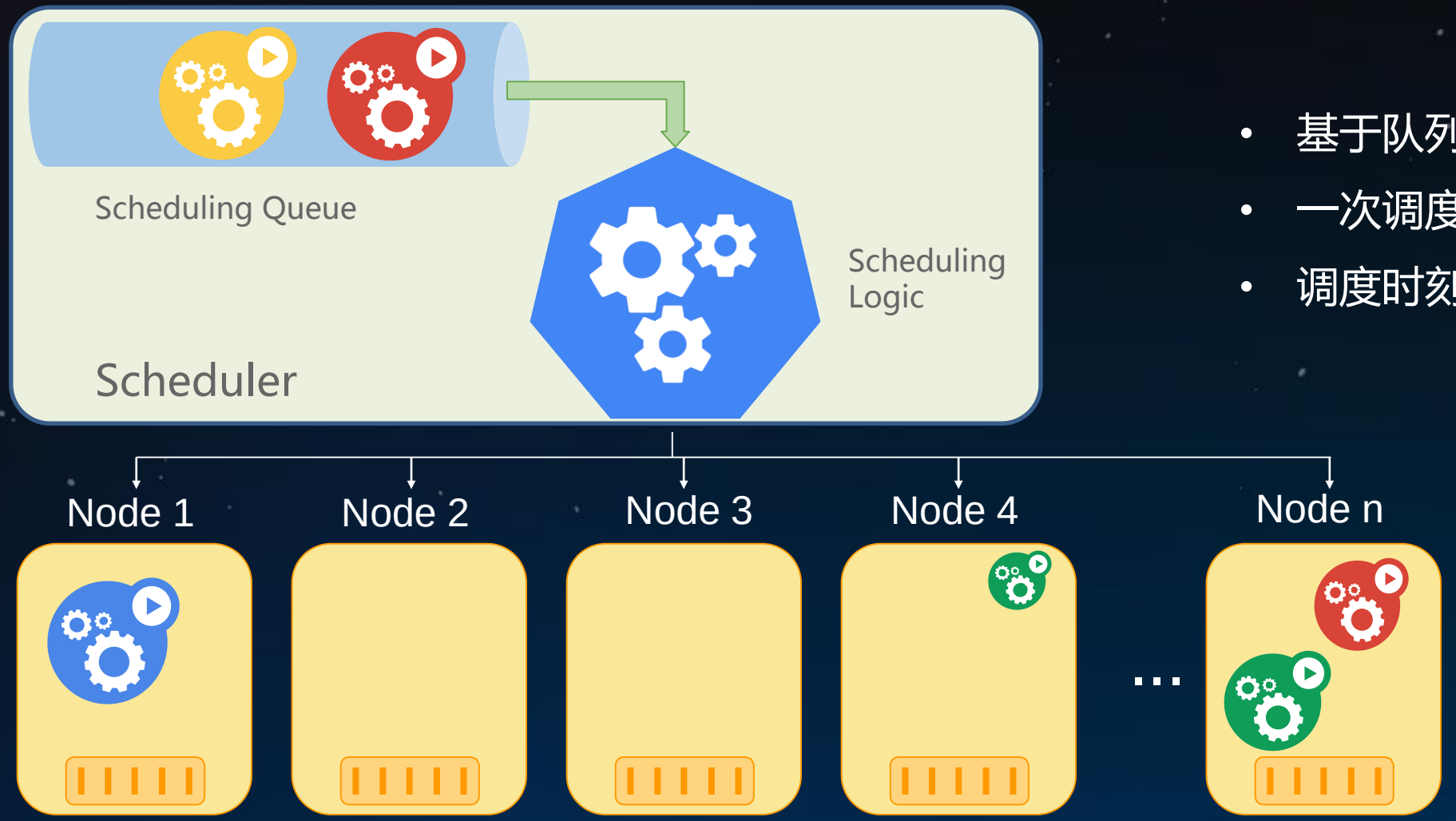
Scheduler：为Pod找到一个合适的Node




```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod-76559f5d5b-l9b9p
......
spec:
  dnsPolicy: ClusterFirst
  nodeName:
  restartPolicy: Always
  containers:
......
```

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-76559f5d5b-l9b9p
......
spec:
  dnsPolicy: ClusterFirst
  nodeName: node1
  restartPolicy: Always
  containers:
......
```

Kubernetes的Default scheduler

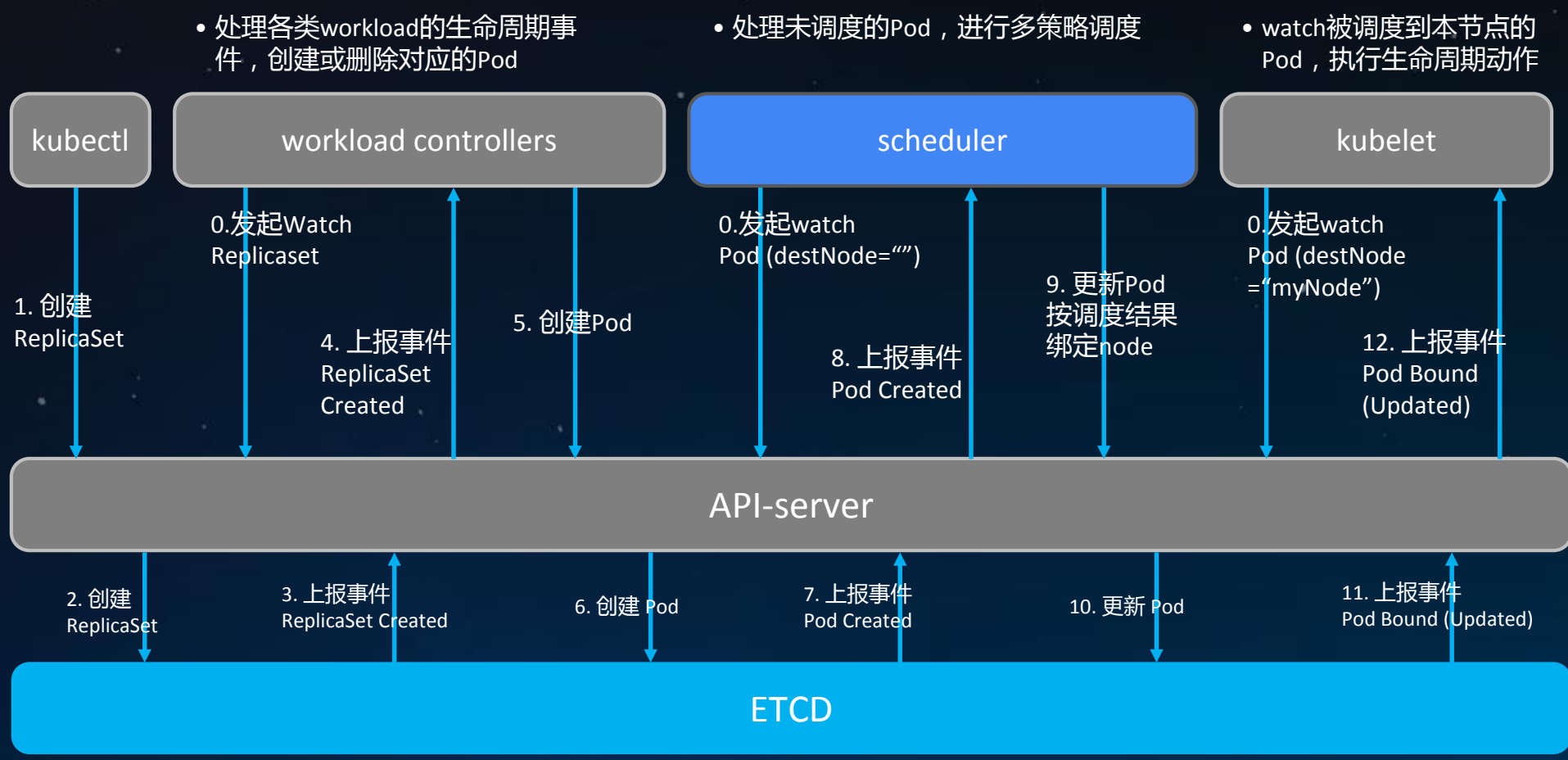


- 基于队列的调度器
- 一次调度一个Pod
- 调度时刻全局最优

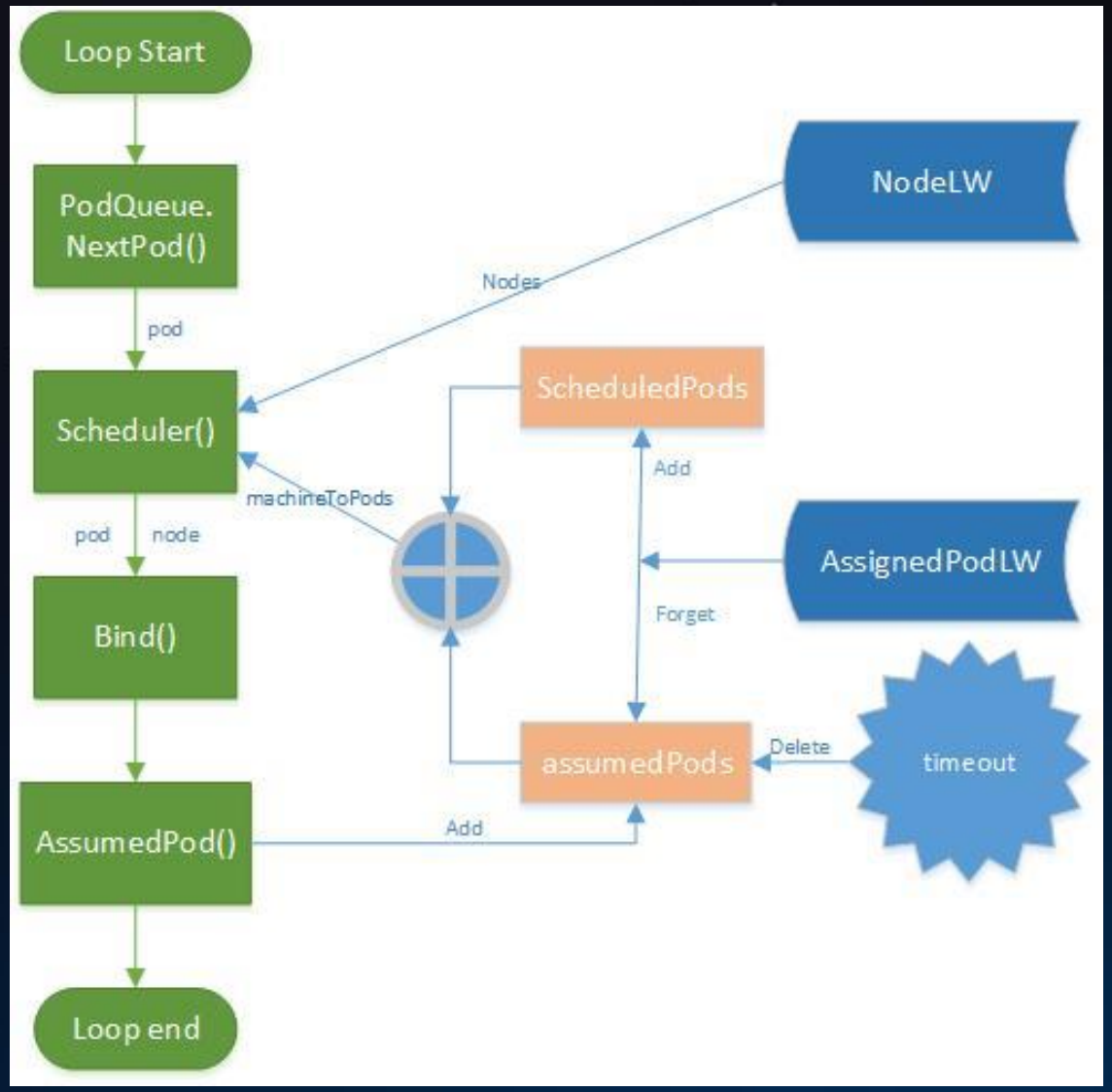
从外部流程看调度器

HUAWEI

从Pod创建开始，到Pod被Bind结束



调度器的内部流程

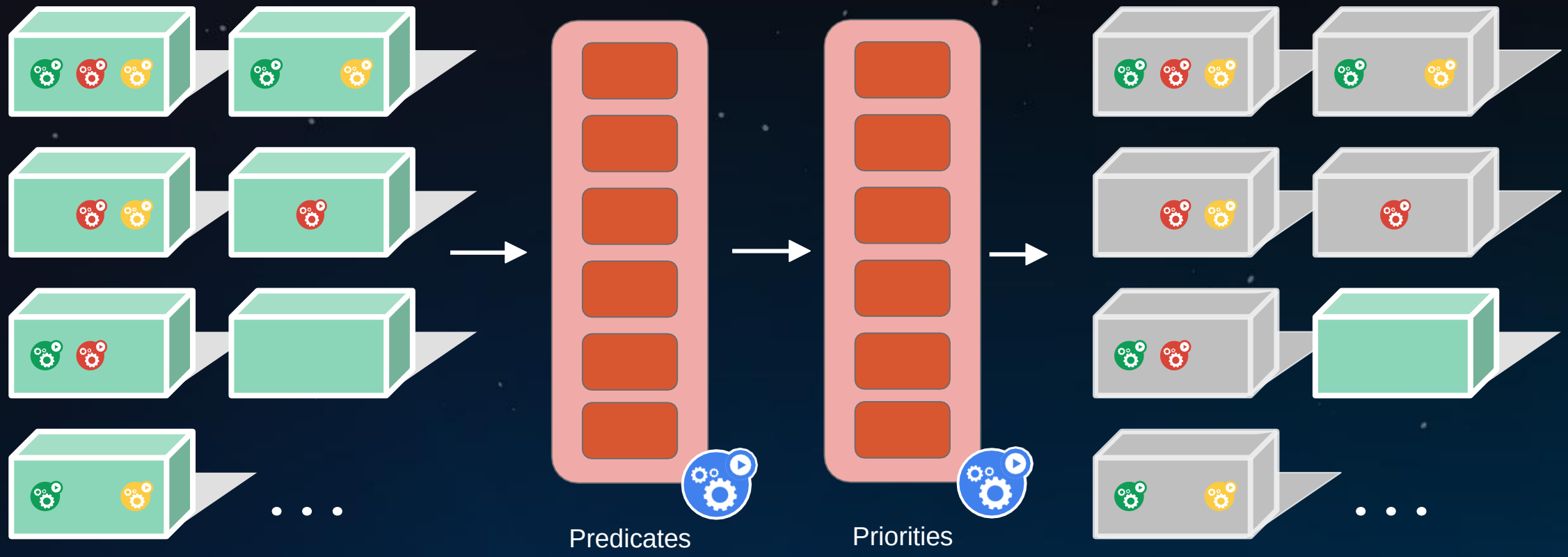


- 通过NodeLister获取所有节点信息；
- 整合scheduled pods和assume pods，合并到pods，作为所有已调度Pod信息；
- 从pods中整理出node-pods的对应关系表nodeNameToInfo；
- 过滤掉不合适的节点；
- 给剩下的节点依次打分；
- 在分数最高的nodes中随机选择一个节点用于绑定。这是为了避免分数最高的节点被几次调度撞车。

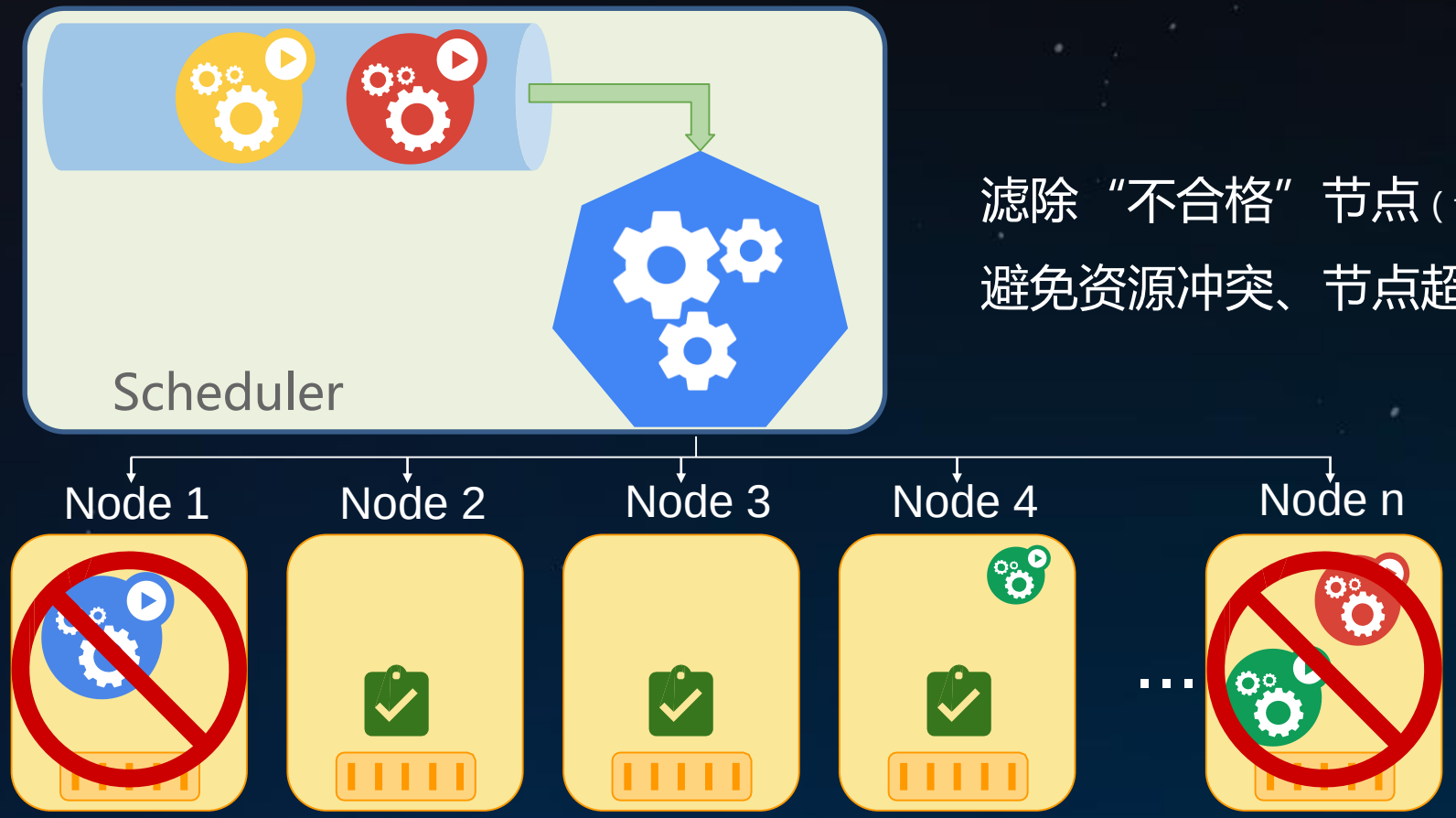
大纲

- K8S调度机制介绍
- **K8S中的调度策略与算法**
- K8S高级调度特性详解

K8S中的调度策略与算法



通过Predicate策略筛选符合条件的Node



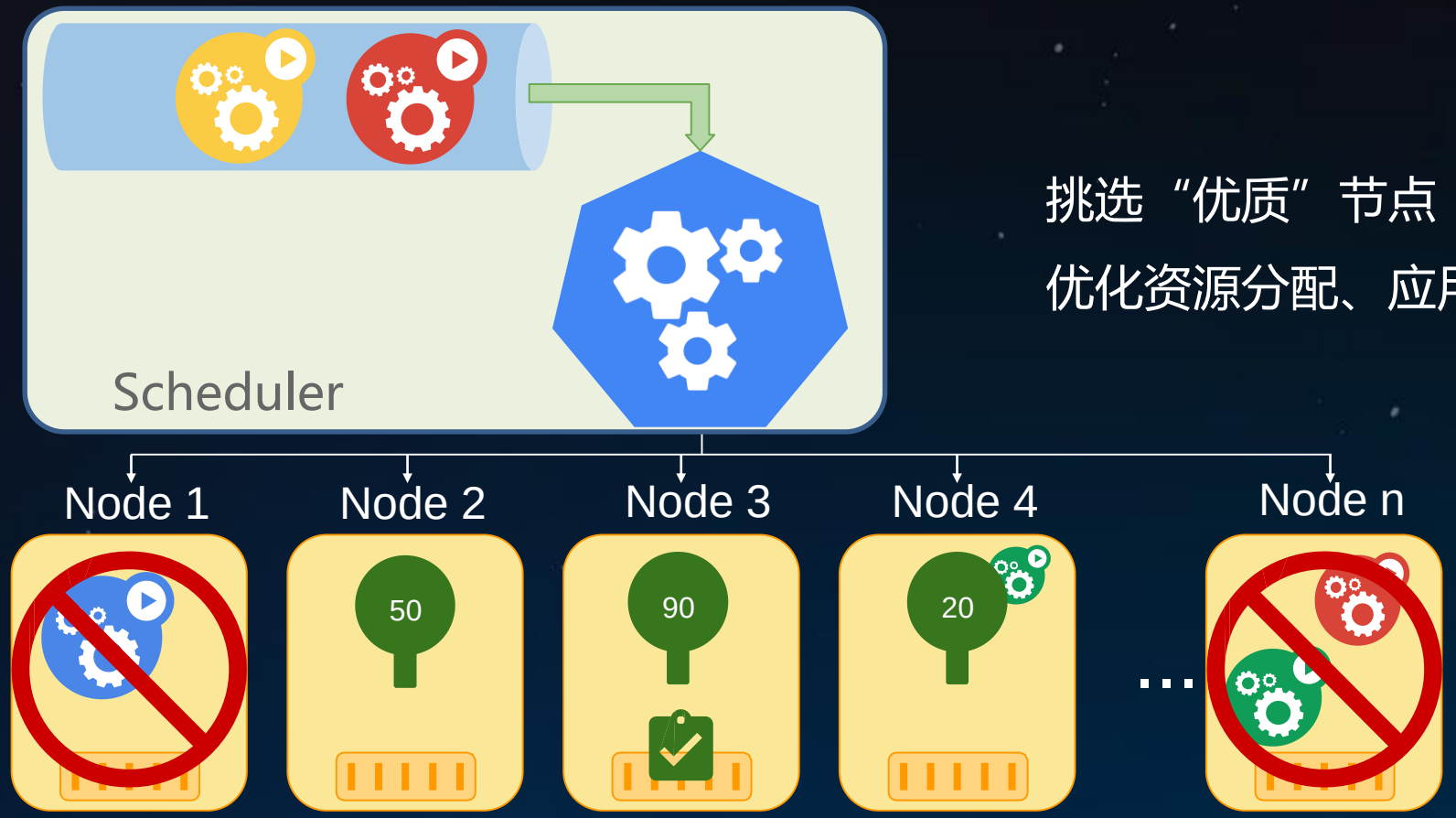
滤除“不合格”节点（调度上去也跑不起来），
避免资源冲突、节点超载等

典型Predicate算法

HUAWEI

算法名称	功能
GeneralPredicates	包含3项基本检查：节点、端口和规则
NoDiskConflict	检查Node是否可以满足Pod对硬盘的需求
NoVolumeZoneConflict	单集群跨AZ部署时，检查node所在的zone是否能满足Pod对硬盘的需求
MaxEBSVolumeCount	部署在AWS时，检查node是否挂载了太多EBS卷
MaxGCEPDVolumeCount	部署在GCE时，检查node是否挂载了太多PD卷
PodToleratesNodeTaints	检查Pod是否能够容忍node上所有的taints
CheckNodeMemoryPressure	当Pod QoS为besteffort时，检查node剩余内存量，排除内存压力过大的node
MatchInterPodAffinity	检查node是否满足pod的亲和性、反亲和性需求

通过Priority策略给剩余的Node评分，挑选最优的节点



挑选“优质”节点，

优化资源分配、应用分布

典型Priority算法

算法名称	功能
LeastRequestedPriority	按node计算资源(CPU/MEM)剩余量排序，挑选最空闲的node
BalancedResourceAllocation	补充LeastRequestedPriority，在cpu和mem的剩余量取平衡
SelectorSpreadPriority	同一个Service/RC下的Pod尽可能的分散在集群中。Node上运行的同个Service/RC下的Pod数目越少，分数越高。
NodeAffinityPriority	按soft(preferred) NodeAffinity规则匹配情况排序，规则命中越多，分数越高
TaintTolerationPriority	按pod tolerations与node taints的匹配情况排序，越多的taints不匹配，分数越低
InterPodAffinityPriority	按soft(preferred) Pod Affinity/Anti-Affinity规则匹配情况排序，规则命中越多，分数越高/低

大纲

- K8S调度机制介绍
- K8S中的调度策略与算法
- **K8S高级调度特性详解**

K8S中的Label与Selector

任意的 metadata

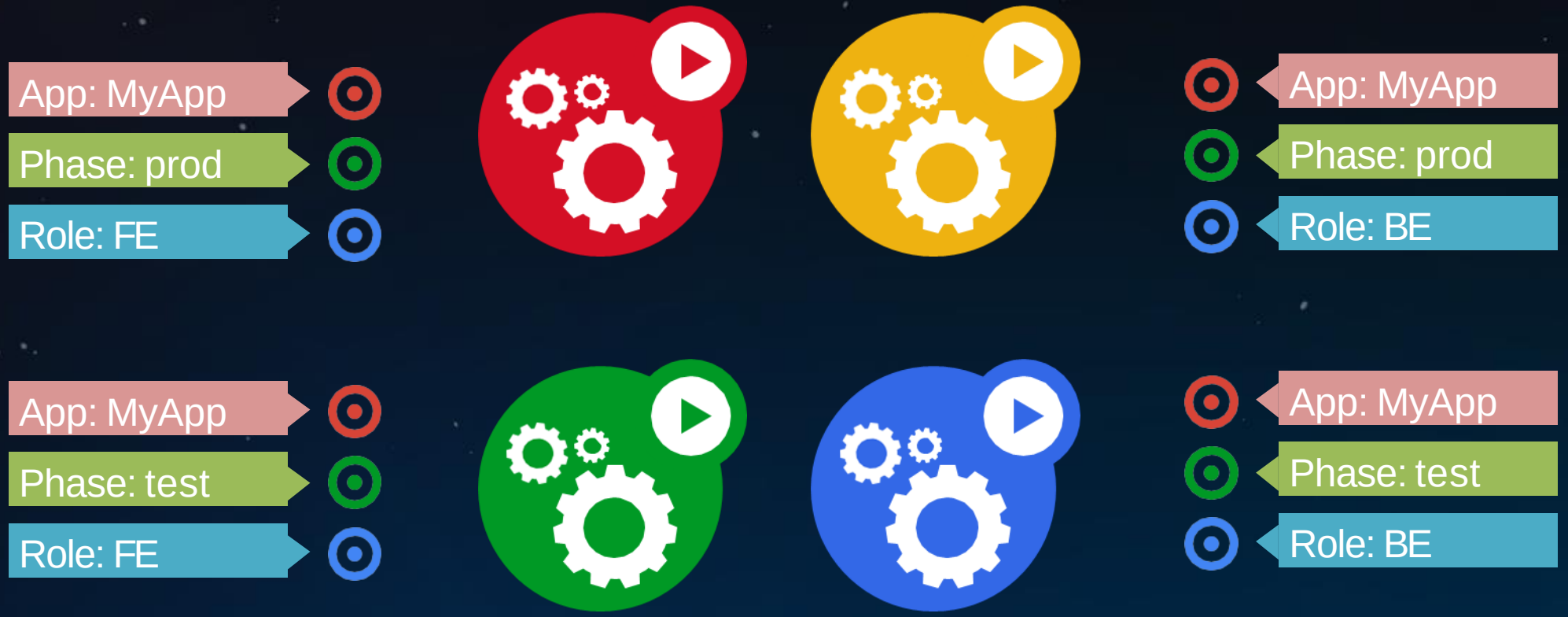
所有API对象都有Label

通常用来标记“身份”

可以查询时用**selectors**过滤

- 类似SQL ‘*select ... where ...*’

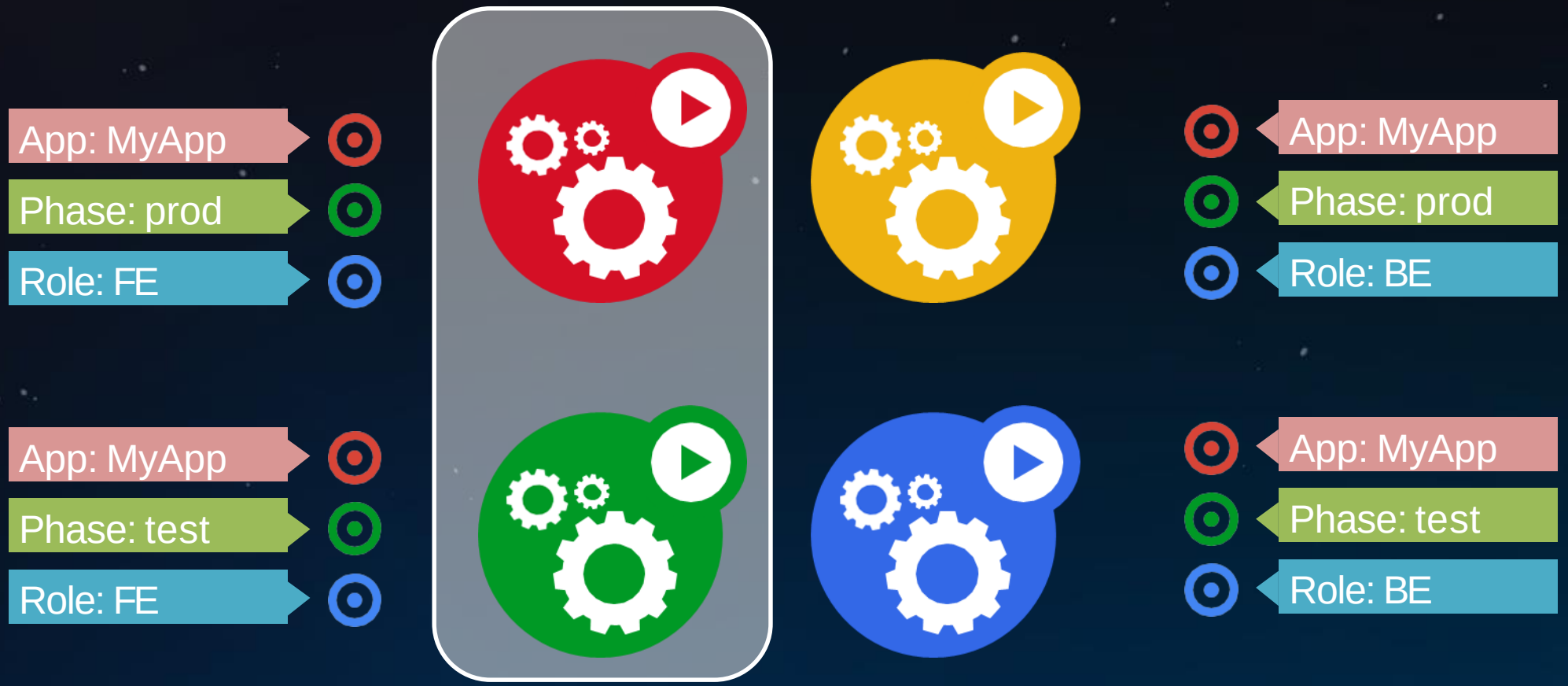
K8S中的Label与Selector



K8S中的Label与Selector

App = MyApp

K8S中的Label与Selector



App = MyApp, Role = FE

Node Affinity 让Pod在一组指定的Node上运行

HUAWEI

Pod Spec

```
nodeAffinity:
  labelSelector: "zone" In
{"central"}
```

nodeSelector升级版：

- 支持In和NotIn两种匹配操作
- 支持匹配相同label-key的多个取值

Node 1

Node 2

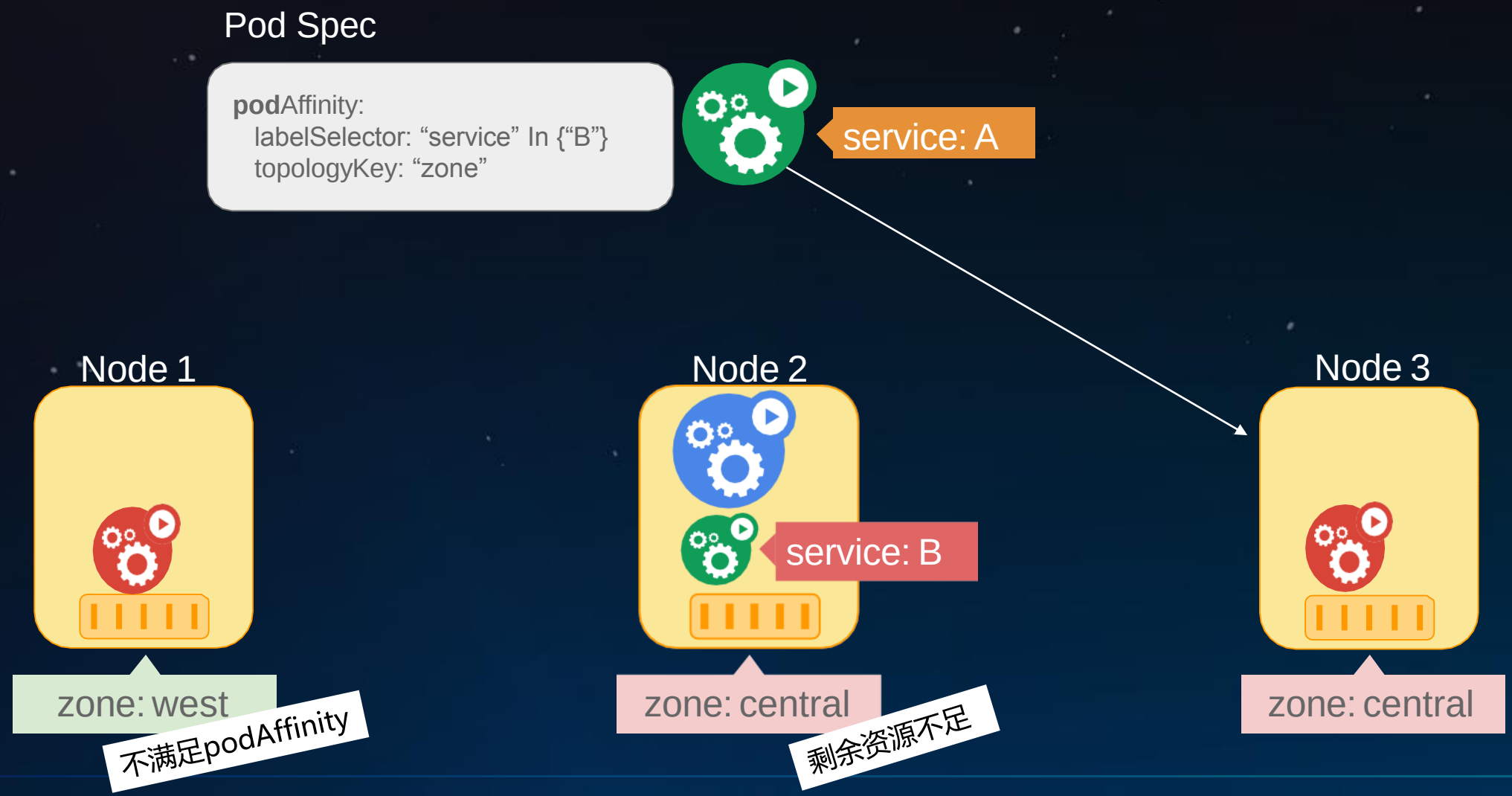
Node 3

zone: west

zone: central

zone: central

Pod Affinity 让Pod与指定Service的一组Pod在相同Node上运行



Pod Affinity 让Pod与指定Service的一组Pod在相同Node上运行

HUAWEI

Pod Spec

```
podAffinity:
  labelSelector: “service” In {“B”}
  topologyKey: “hostname”
```

service: A

Pod is not schedulable

Node 1

zone: west

不满足podAffinity

Node 2

service: B

zone: central

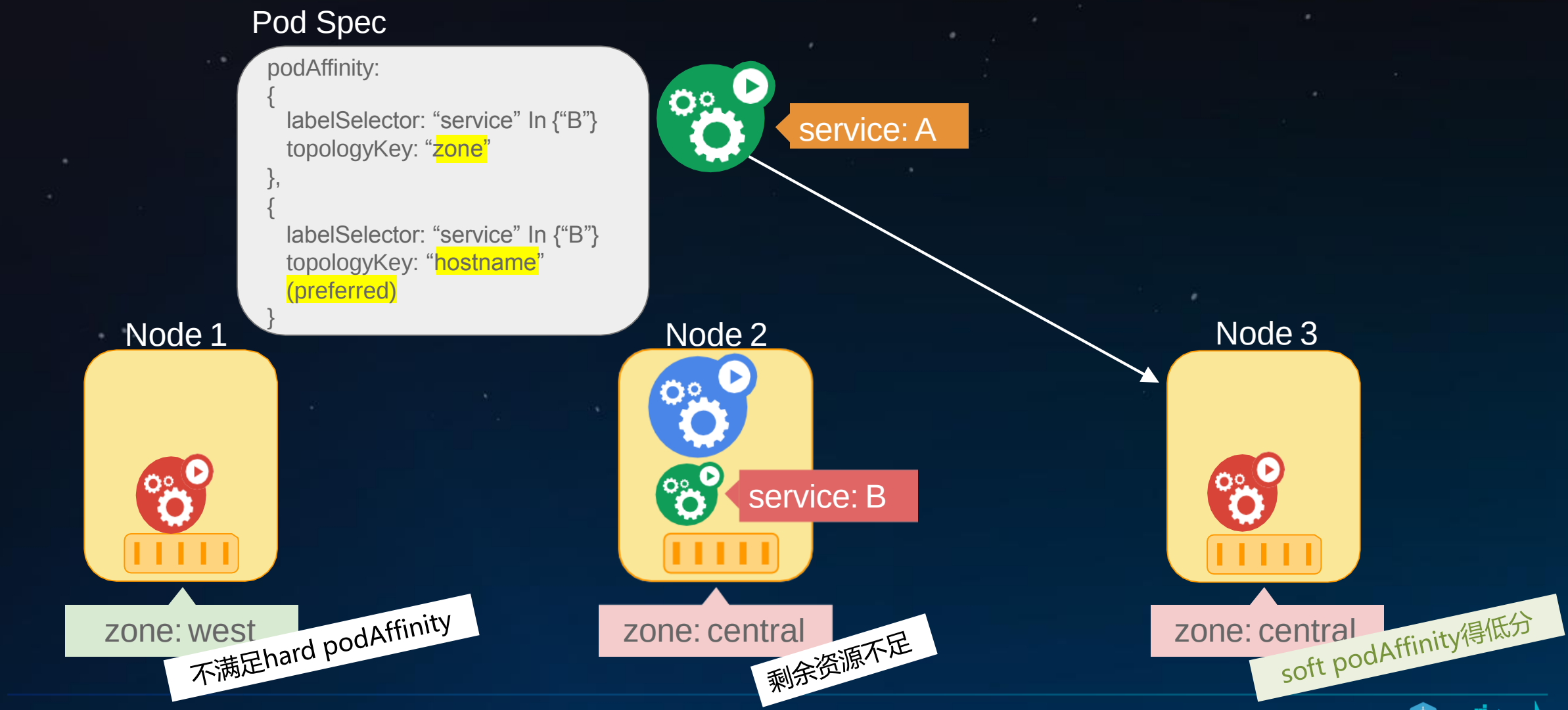
剩余资源不足

Node 3

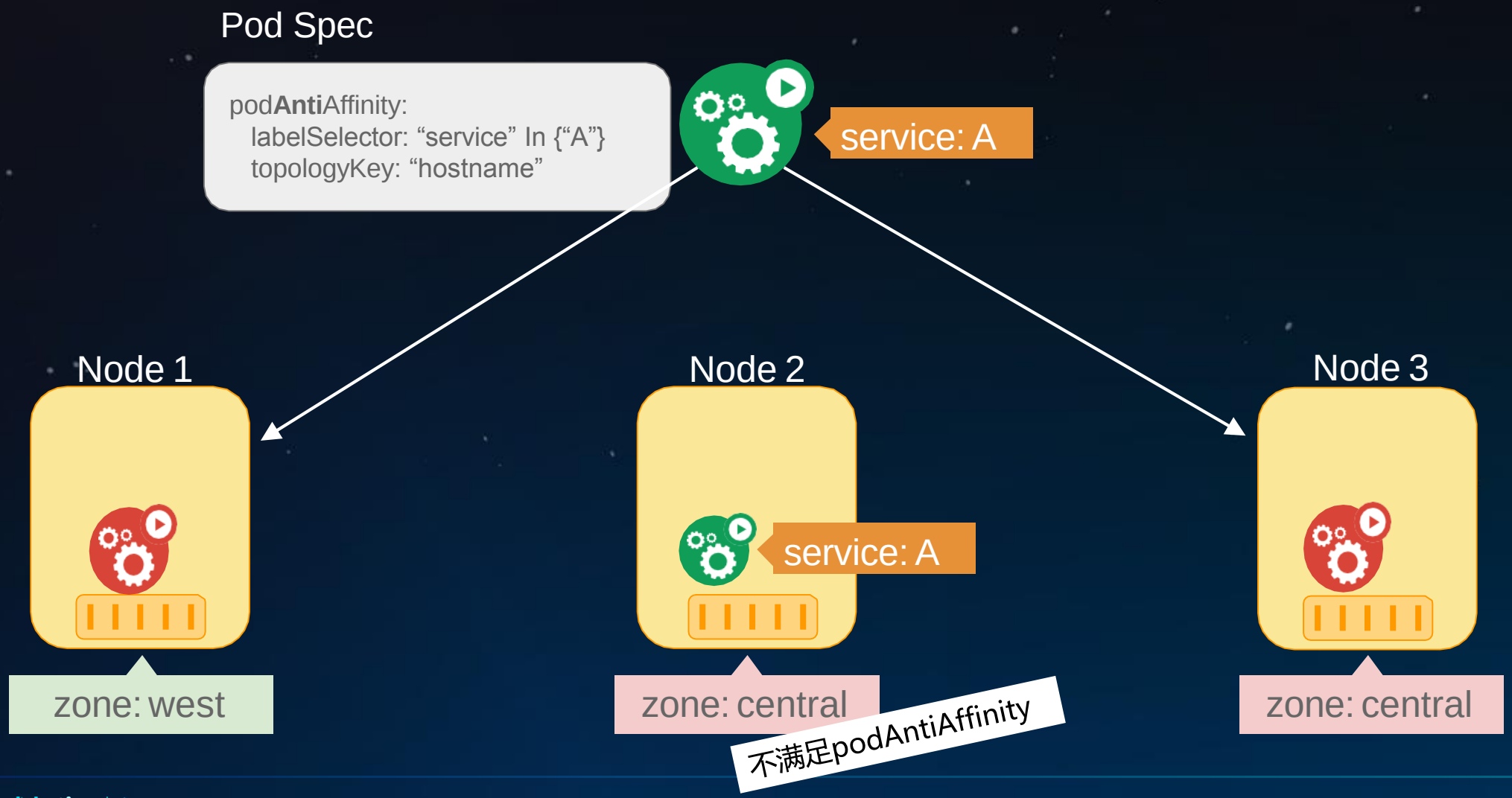
zone: central

不满足podAffinity

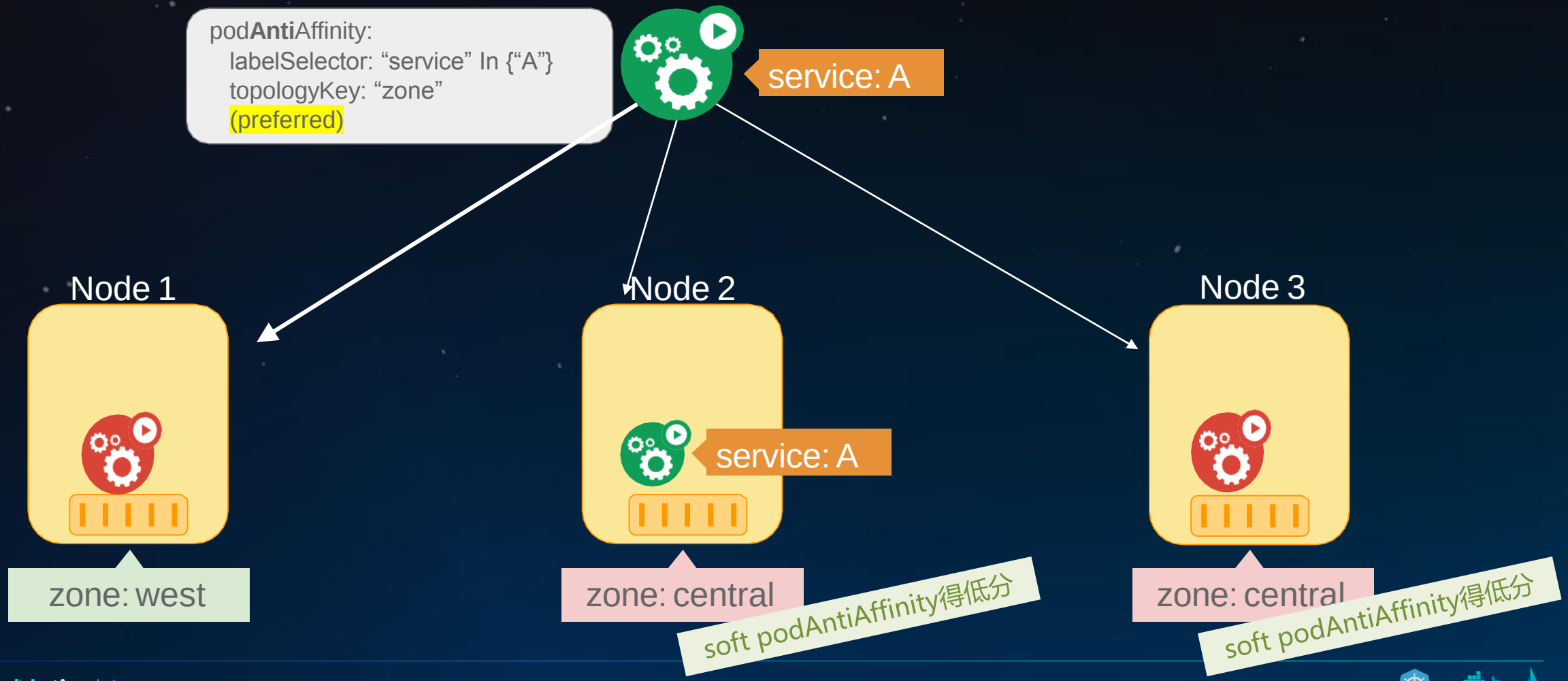
Pod Affinity让Pod与指定Service的一组Pod在相同Node上运行



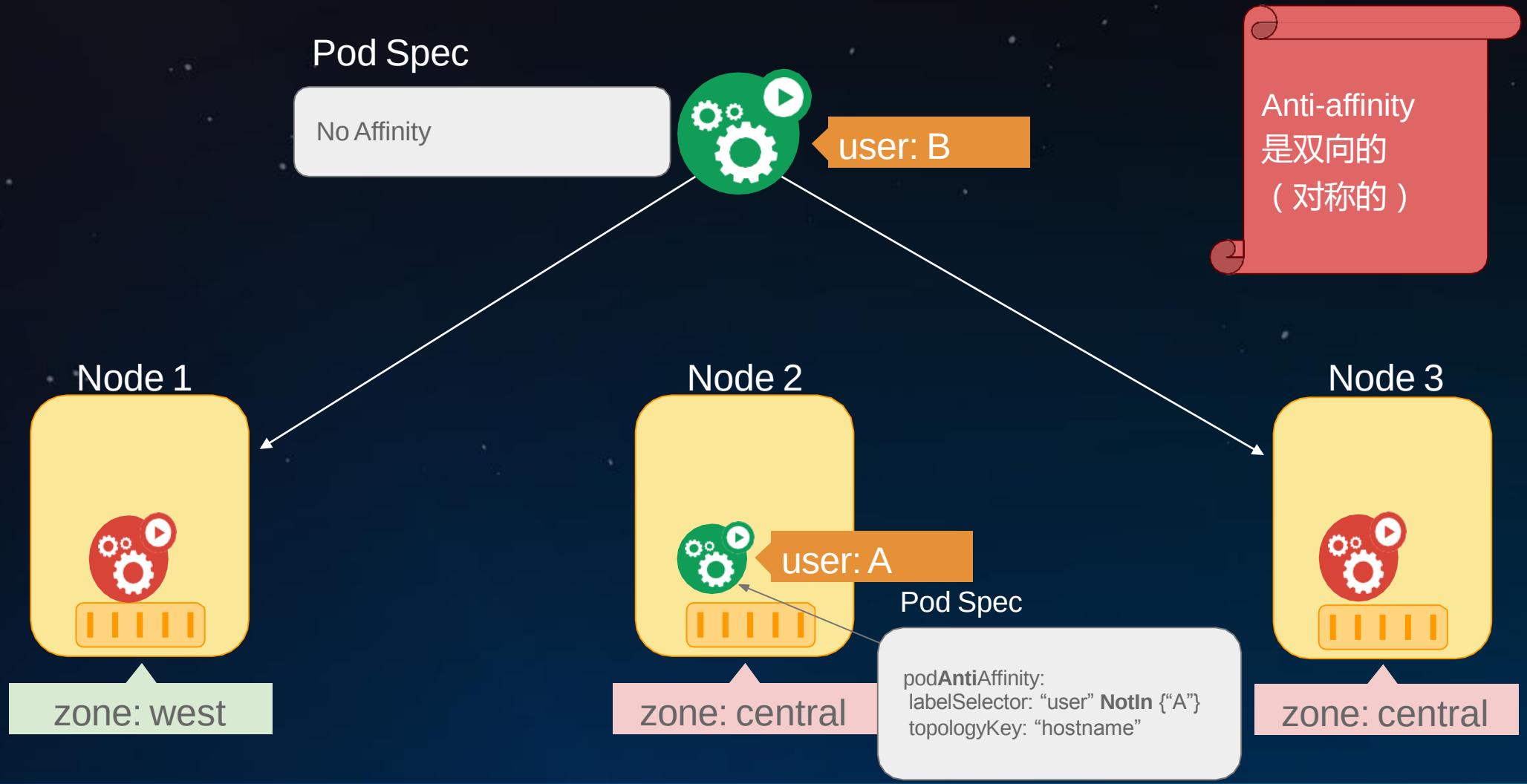
Pod Anti-Affinity 让同一个Service的Pod分在到不同Node上运行



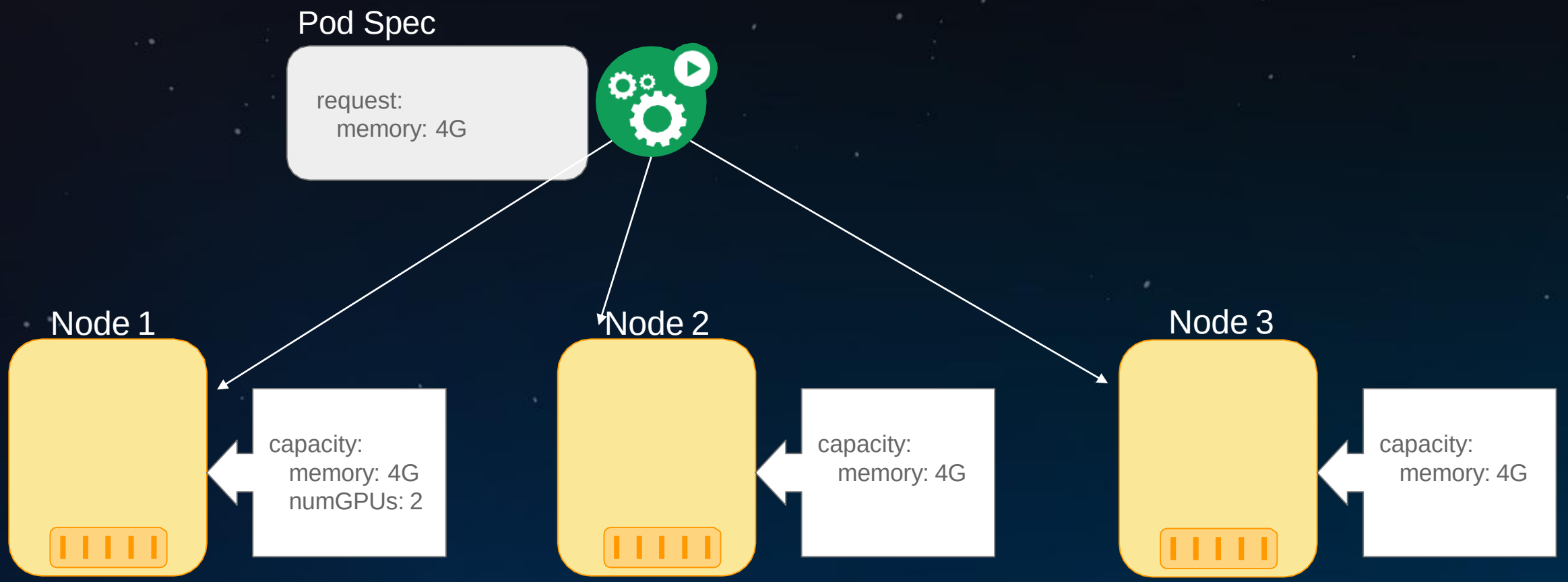
Pod Anti-Affinity 让同一个Service的Pod分在到不同Node上运行



Pod Anti-Affinity具有对称性

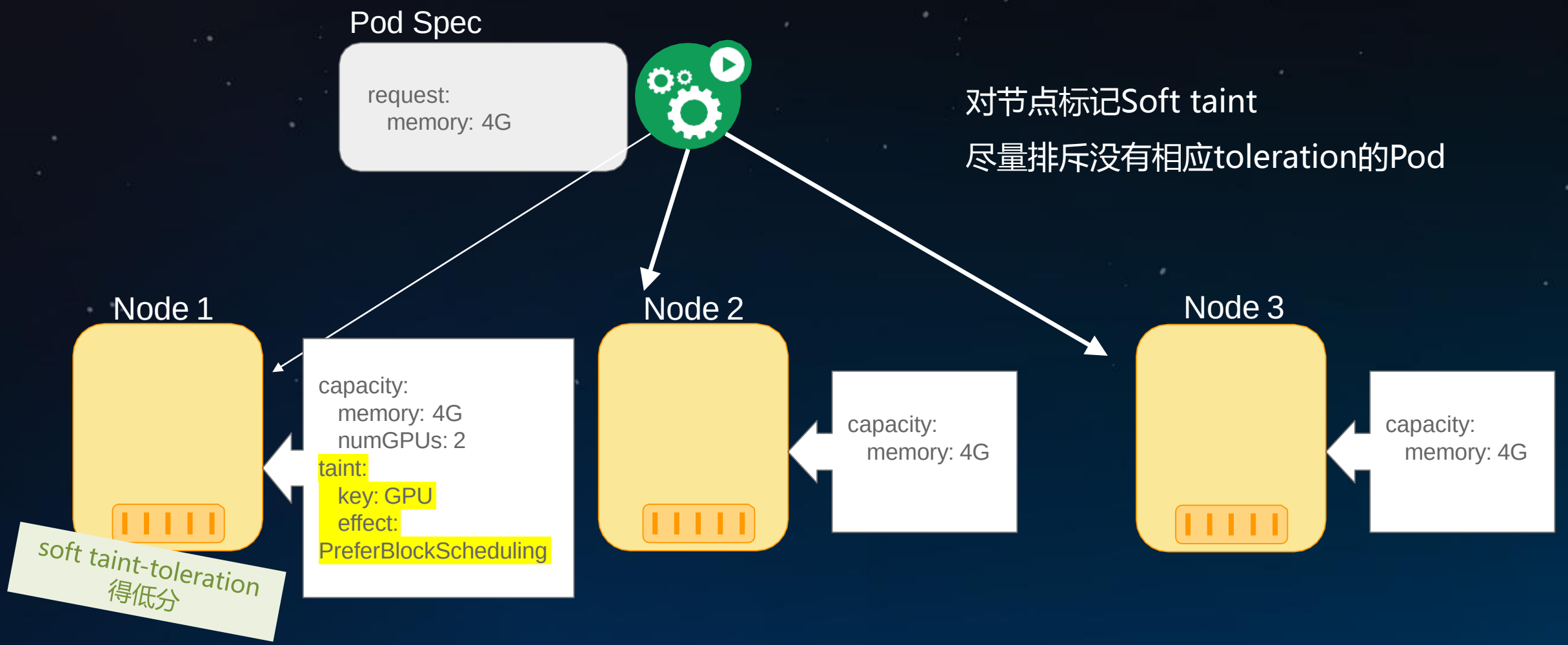


Taints-tolerations 来自Node的反亲和配置



Taints-tolerations 来自Node的反亲和配置

Taints-tolerations 来自Node的反亲和配置



Taints-tolerations 来自Node的反亲和配置

Taints-tolerations 来自Node的反亲和配置

或者对节点标记Hard taint

强行阻止没有相应toleration的Pod

HUAWEI

Thank You

http://zhibo.huaweicloud.com/watch/2174406

直播 每周四 晚20:00