62Kubernetes 系列（五十七）Kubernetes 中的 Pod 重平衡和分配

Kubernetes 系列（五十七）Kubernetes 中的 Pod 重平衡和分配

欢迎关注我的公众号" **云原生拓展** "，原创技术文章第一时间推送。

Kubernetes 会重新平衡您的 Pod 吗?

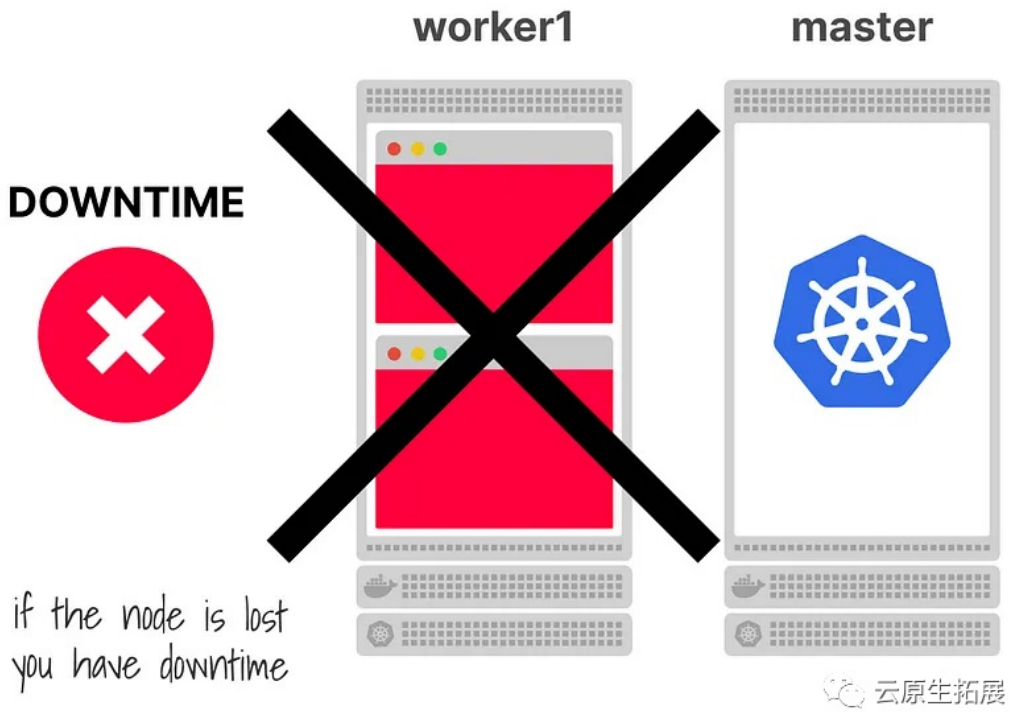
如果有一个节点有更多空间，Kubernetes 是否会重新计算并平衡工作负载?

让我们看一个例子。

您有一个包含单个节点的集群，该节点可以托管 2 个 Pod。

如果节点崩溃，您将遇到停机时间。

你可以有第二个节点，每个节点有一个 Pod 来防止这种情况。

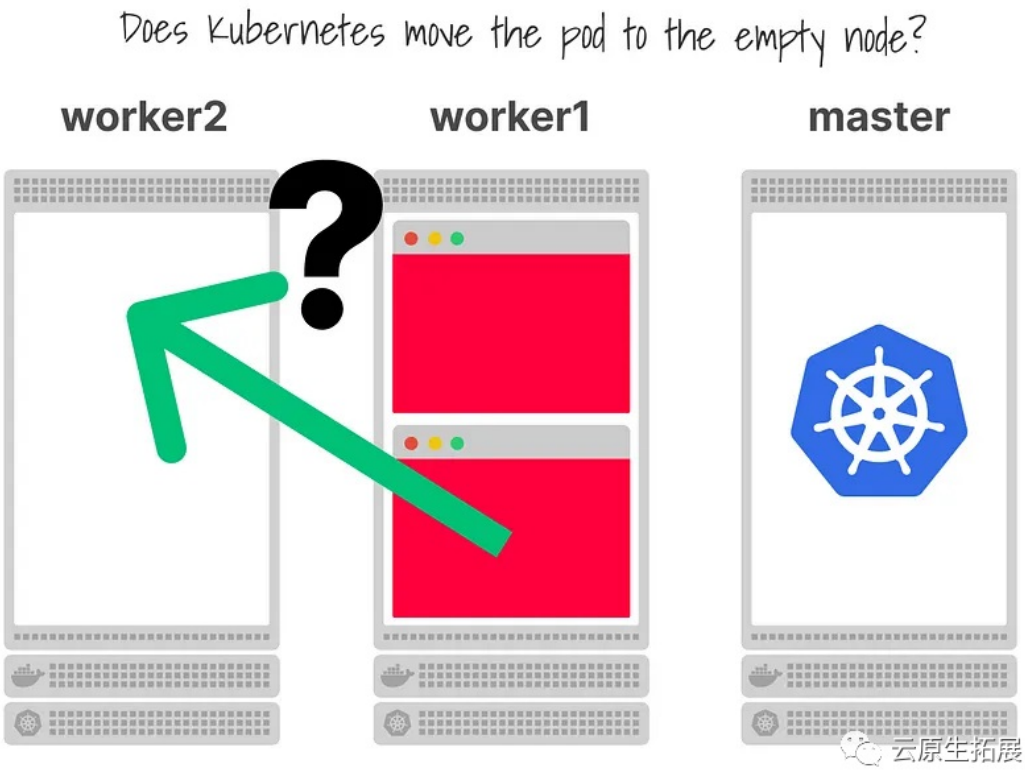


您供应第二个节点。

接下来发生什么?

Kubernetes 是否注意到您的 Pod 有空间?

它会移动第二个 Pod 并重新平衡集群吗?



不幸的是，事实并非如此。

但是为什么?

定义 Deployment 时，您指定:

• Pod 的 template。

• 份数（replicas）。

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
```

pod definition

但是在那个文件中没有任何地方你说你想要每个节点一个副本!

ReplicaSet 计数 2 个 Pod，并且与所需状态匹配。

Kubernetes 不会采取任何进一步的行动。

换句话说，Kubernetes 不会自动重新平衡你的 pod。

但你可以解决这个问题。

共有三种流行的选择：

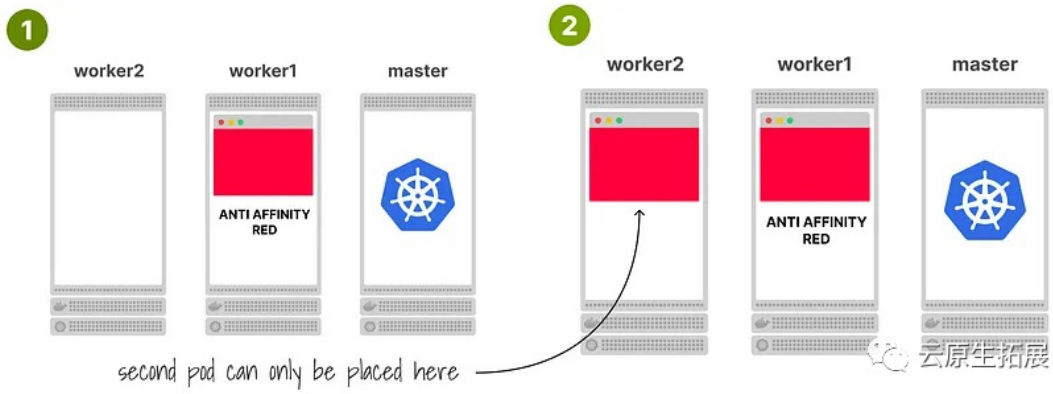
1. Pod（反）亲和力。

2. Pod 拓扑分布约束。

3. Descheduler.

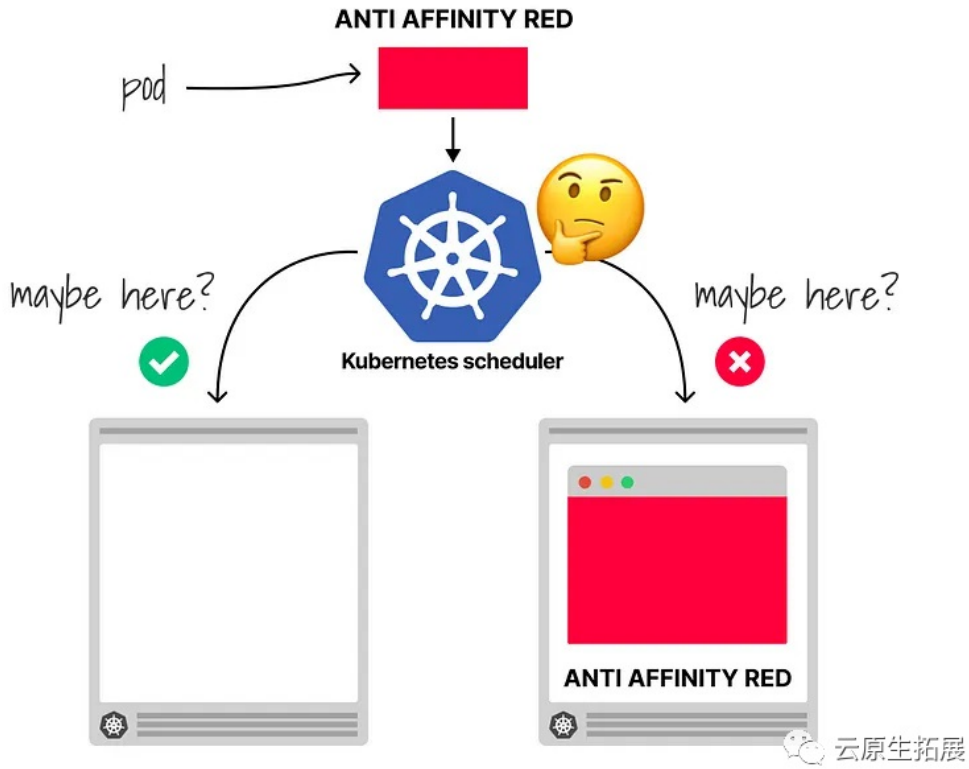
第一个选项是使用 pod 反亲和性。

使用 Pod 反亲和性，您的 Pod 排斥具有相同标签的其他 Pod，迫使它们位于不同的节点上。



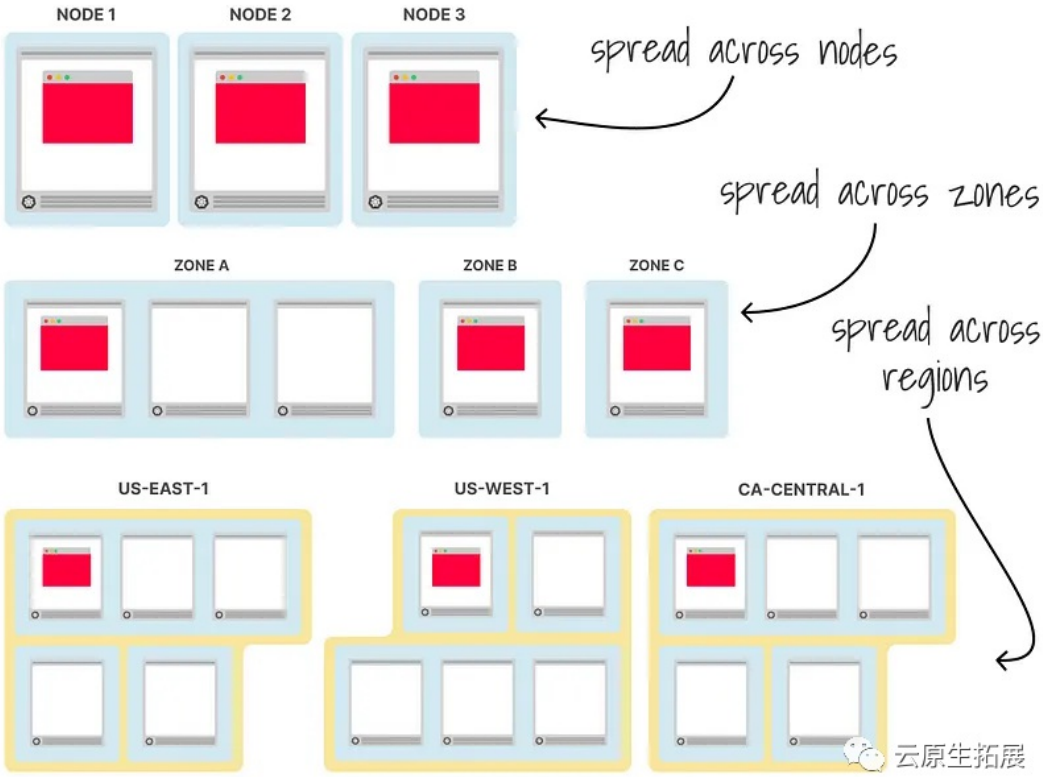
请注意当调度程序分配 pod 时如何评估 pod 亲和力。

它不会追溯应用，因此您可能需要删除一些 pod 以强制调度程序重新计算分配。

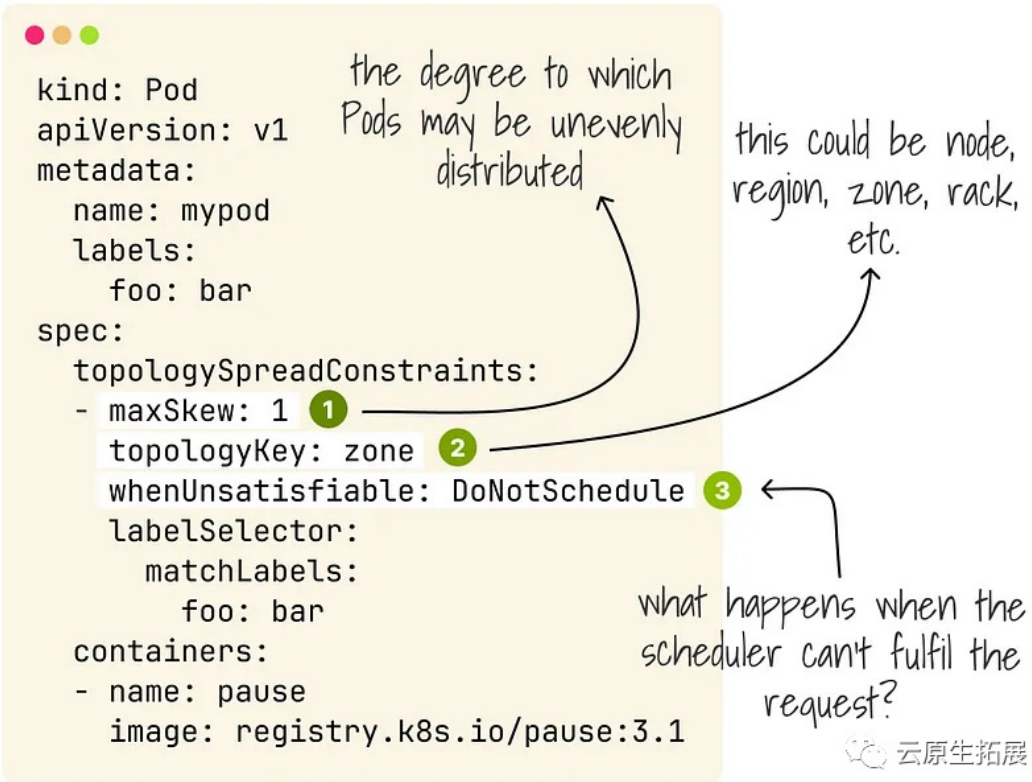


或者，您可以使用拓扑分布约束来控制 Pod 如何在集群中分布在故障域（例如区域、区域、节点等）之间。

这类似于 pod affinity 但更强大。

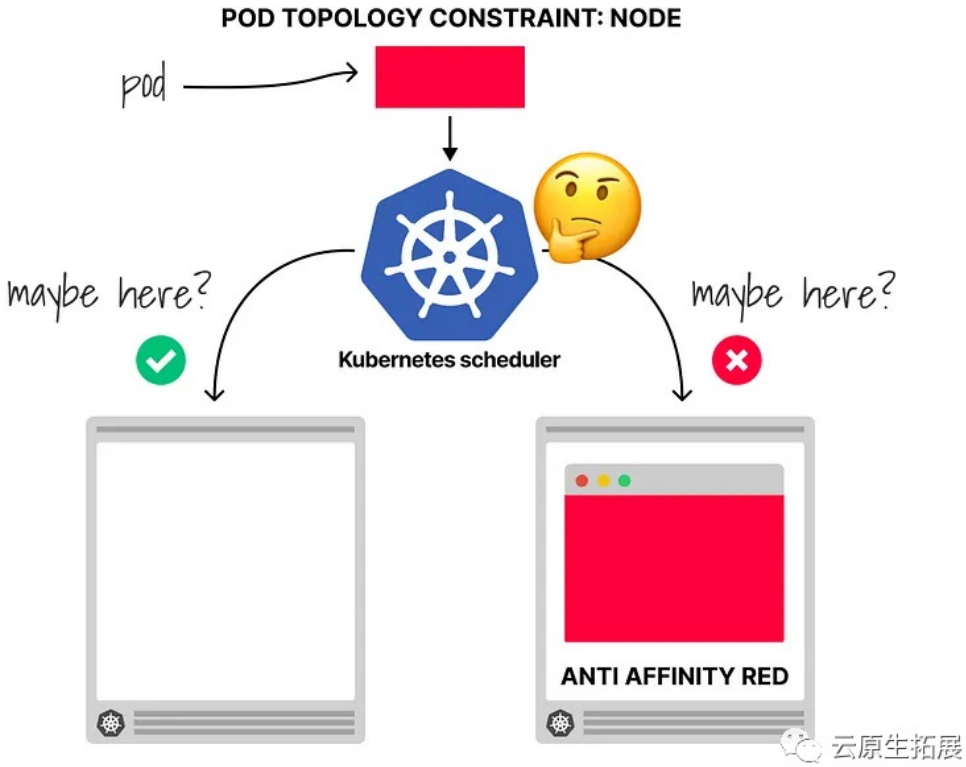


使用拓扑传播约束，您可以选择拓扑并选择 pod 分布（偏斜），当约束无法满足时会发生什么（无论如何调度与不调度）以及与 pod 亲和力和污点的交互。



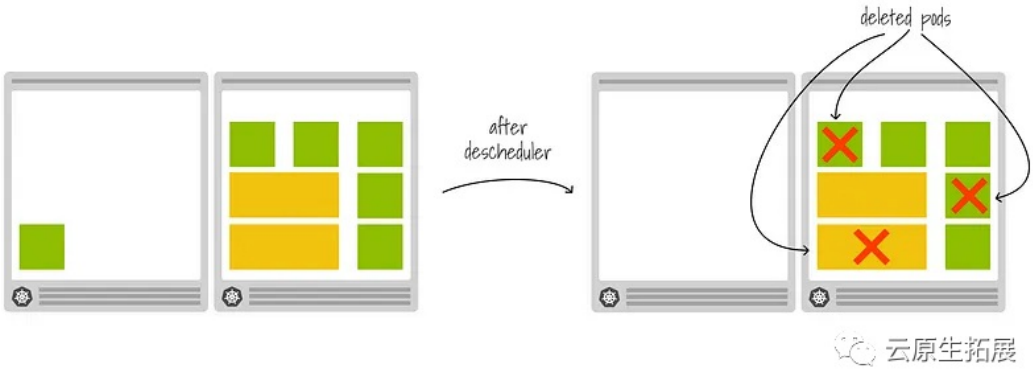
然而，即使在这种情况下，调度程序也会在分配 pod 时评估拓扑扩展约束。

它不适用于追溯——您仍然可以删除 pod 并强制调度程序重新分配它们。



如果你想动态地重新平衡你的 pod（不仅仅是当调度程序分配它们时），你应该检查 Descheduler。

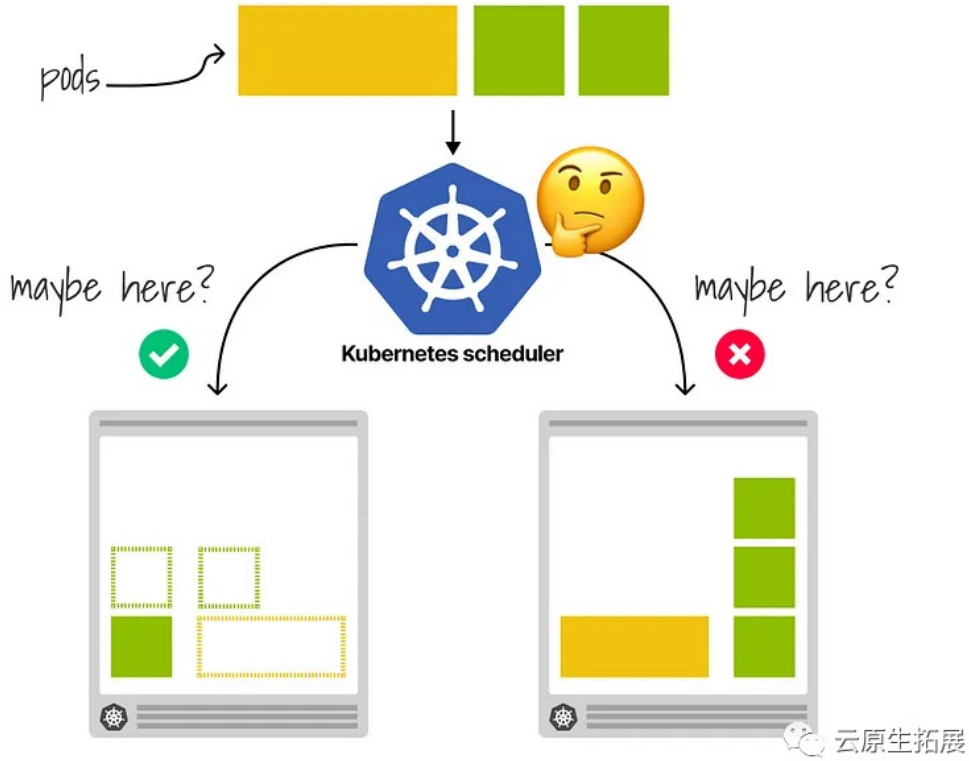
Descheduler 定期扫描您的集群，如果它发现一个节点比其他节点利用率更高，它会删除该节点中的一个 pod。



删除 Pod 时会发生什么?

ReplicaSet 将创建一个新的 Pod，调度程序可能会将其放置在利用率较低的节点中。

如果您的 pod 具有拓扑分布约束或 pod 亲和性，它将相应地分配。



Descheduler 可以根据以下策略驱逐 pod：

• Node 利用率

• Pod 年龄.

• Failed pods.

• Duplicates.

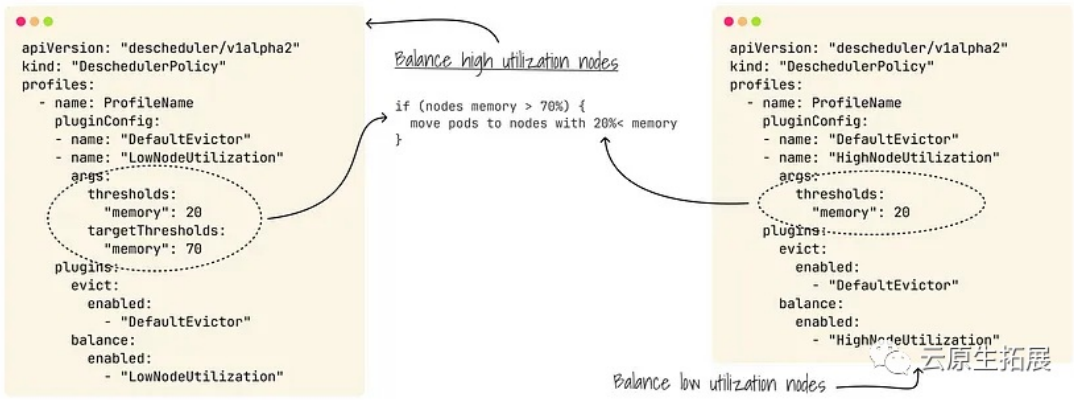
• 亲和性或污点违规。

Evictor Plugin configuration

Name	type	Default Value	Description
`nodeSelector`	`string`	`nil`	limiting the nodes which are processed
`evictLocalStoragePods`	`bool`	`false`	allows eviction of pods with local storage
`evictSystemCriticalPods`	`bool`	`false`	[Warning: Will evict Kubernetes system pods] allows eviction of pods with any priority, including system pods like kube-dns
`ignorePvcPods`	`bool`	`false`	set whether PVC pods should be evicted or ignored
`evictFailedBarePods`	`bool`	`false`	allow eviction of pods without owner references and in failed phase
`labelSelector`	`metav1.LabelSelector`		(see label filtering)
`priorityThreshold`	`priorityThreshold`		(see priority filtering)
`nodeFit`	`bool`	`false`	(see node fit filtering)

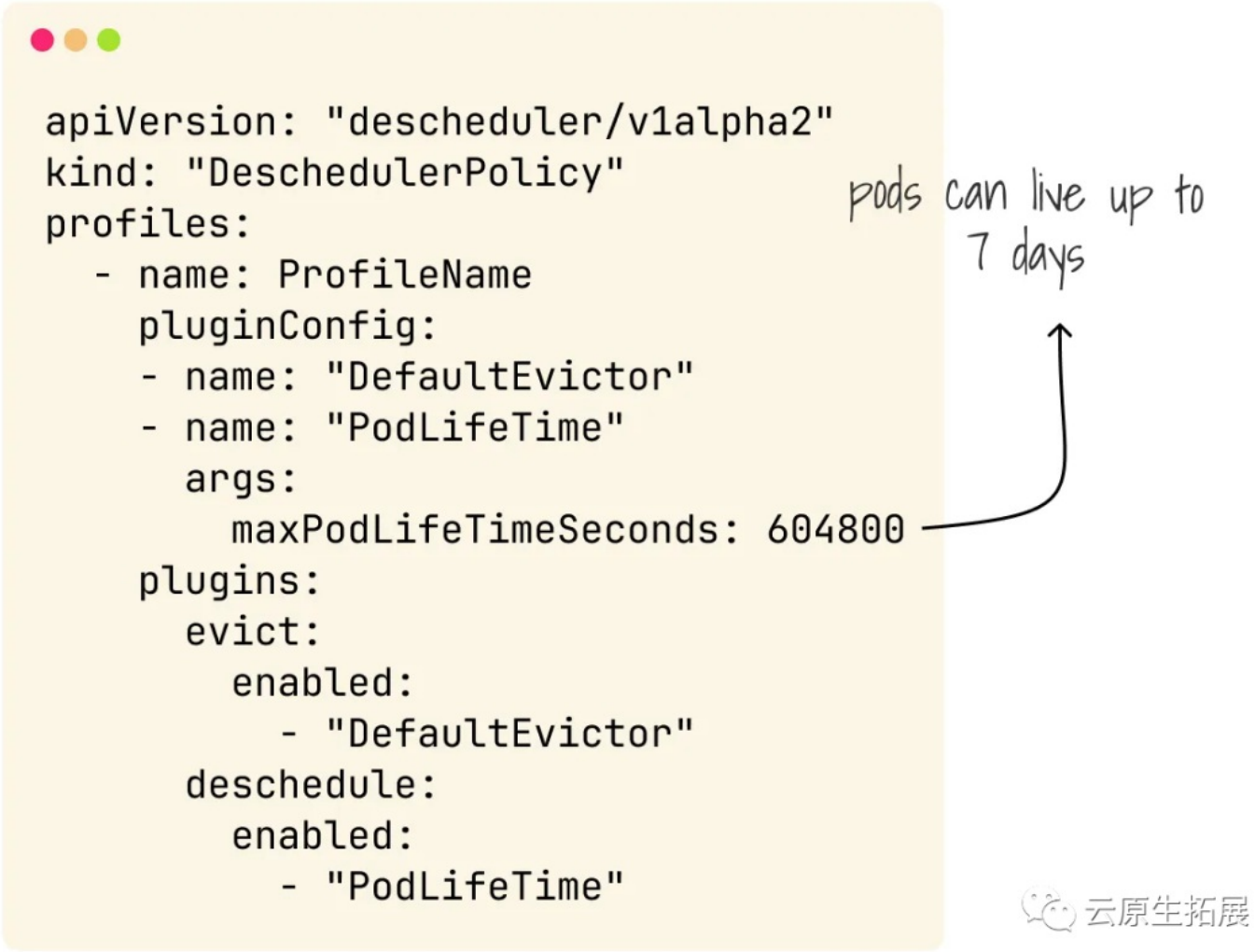
如果您的集群运行时间较长，则资源利用率可能会更加平衡。

以下两种策略可用于根据 CPU、内存或 pod 数量重新平衡集群。



另一项实用政策是防止开发人员和运营商将 pod 视为虚拟机。

您可以使用 descheduler 来确保 pod 仅运行固定时间（例如 7 天）。



最后，您可以将 Descheduler 与 Node Problem Detector 和 Cluster Autoscaler 结合使用，以自动删除有问题的节点。

Descheduler 是控制集群效率的绝佳选择，但默认情况下并未安装。

它可以部署为 Job、CronJob 或 Deployment （https://github.com/kubernetes-sigs/descheduler）