65Kubernetes 系列（六十）为什么无法 ping 通 Kubernetes Service

Kubernetes 系列（六十）为什么无法 ping 通 Kubernetes Service

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。

> 在本文中，您将了解 ClusterIP 服务和 kube-proxy 如何在 Kubernetes 中工作。

你试过在 Kubernetes 中 ping Service IP 地址吗？

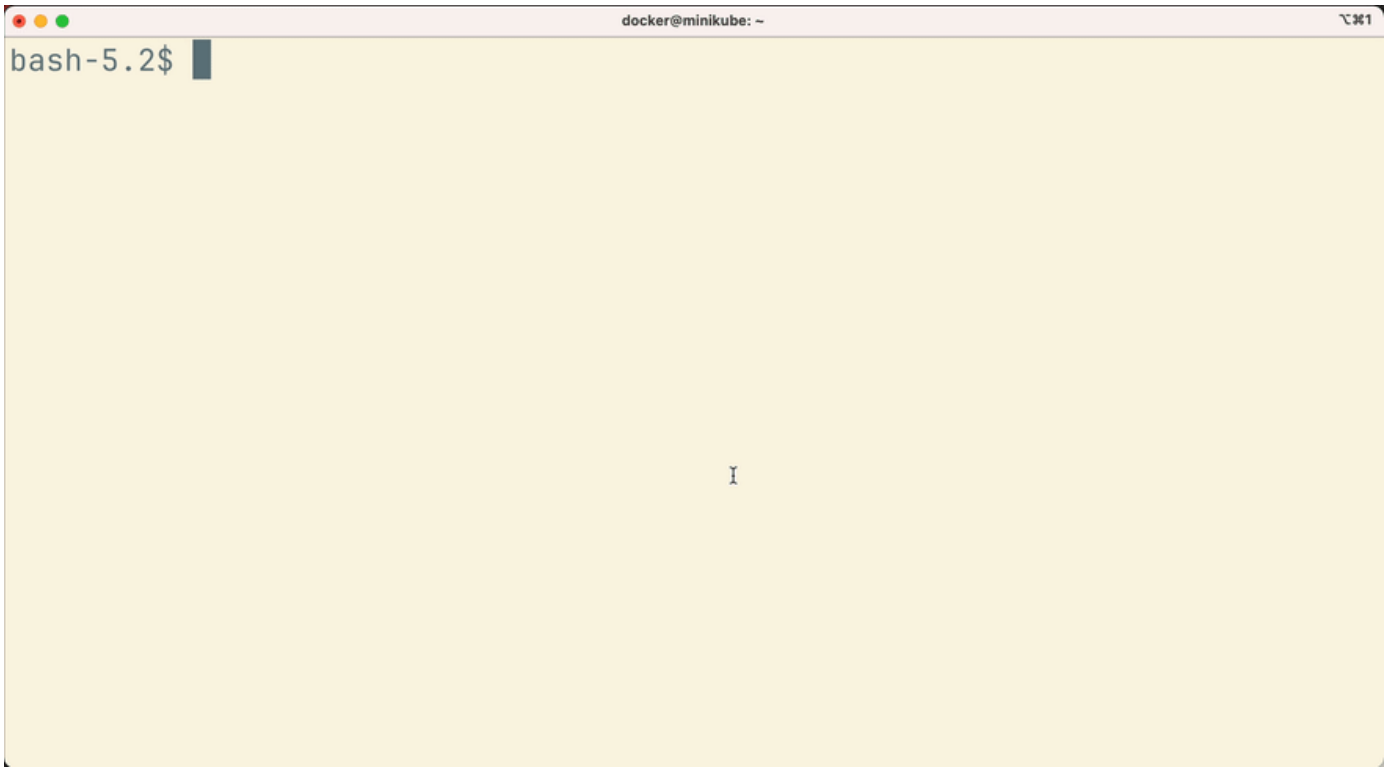
你可能已经注意到它不起作用。

我知道这很令人困惑——让我解释一下。

Kubernetes Service 仅存在于 etcd 中。

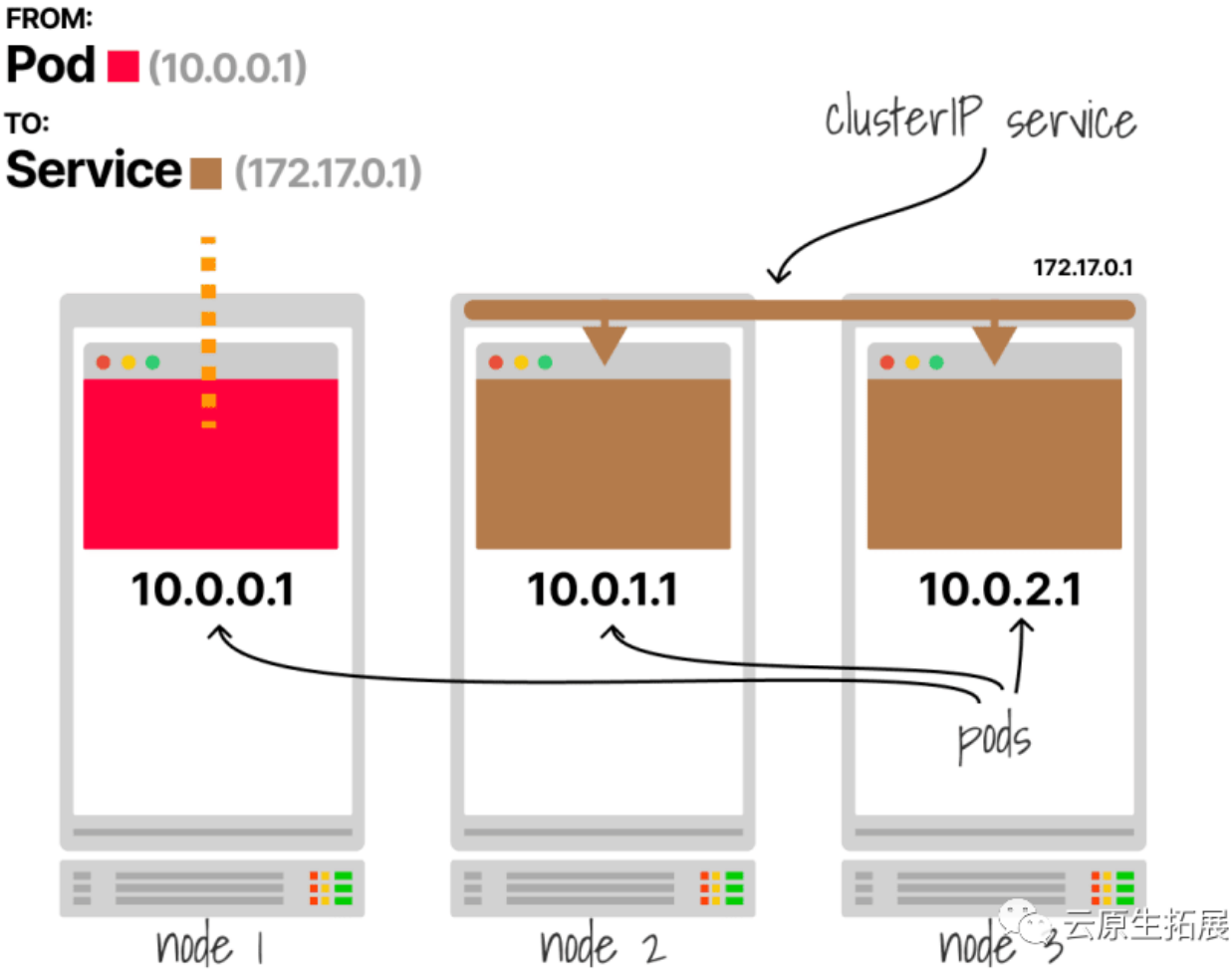
没有进程监听 Service 的 IP 地址和端口。

尝试在节点中执行 `netstat -ntlp` — 什么也没有。



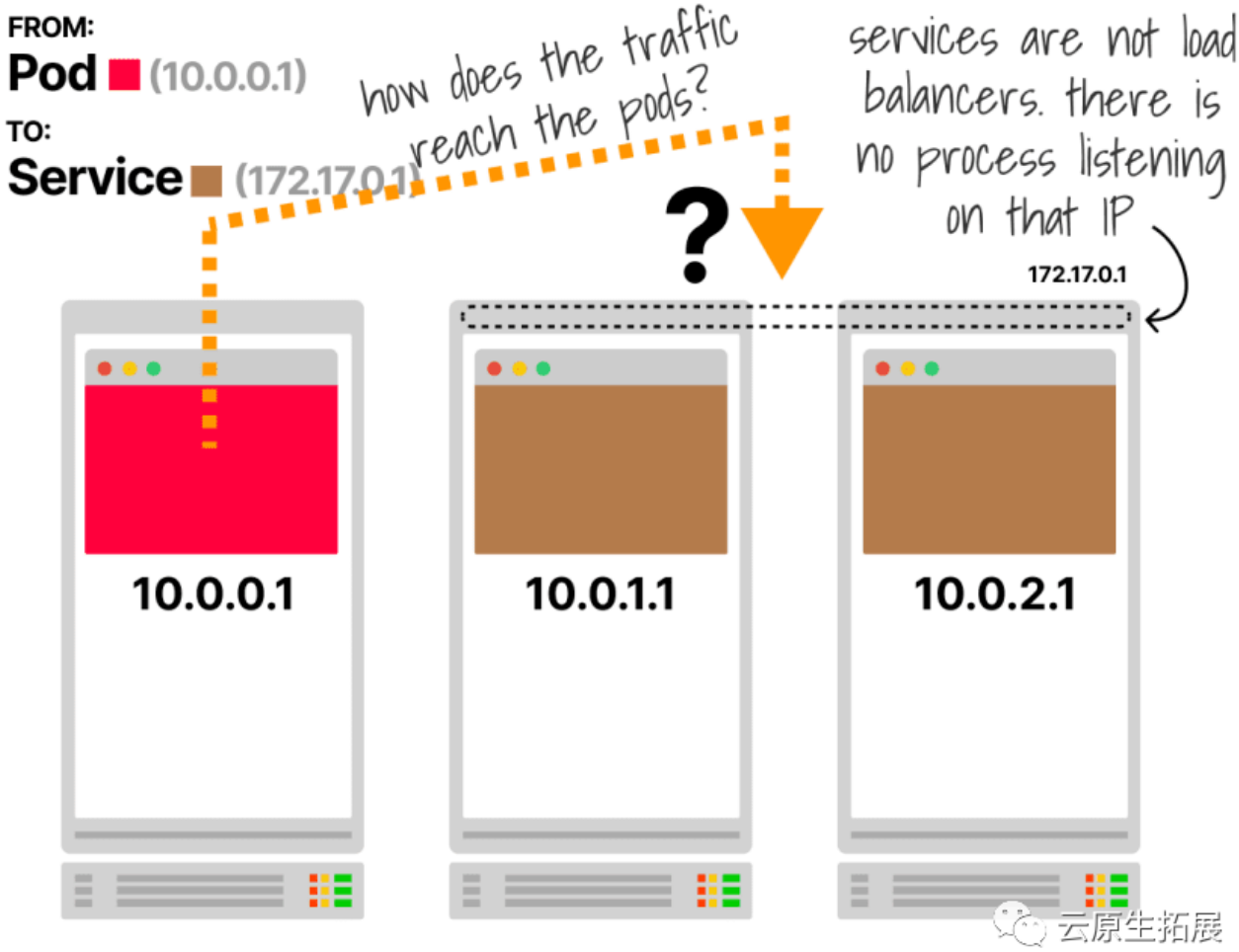
那么它们是如何工作的呢？

考虑一个具有三个节点的集群。 红色 pod 使用 IP `172.17.0.1` 向棕色 Service 发出请求。



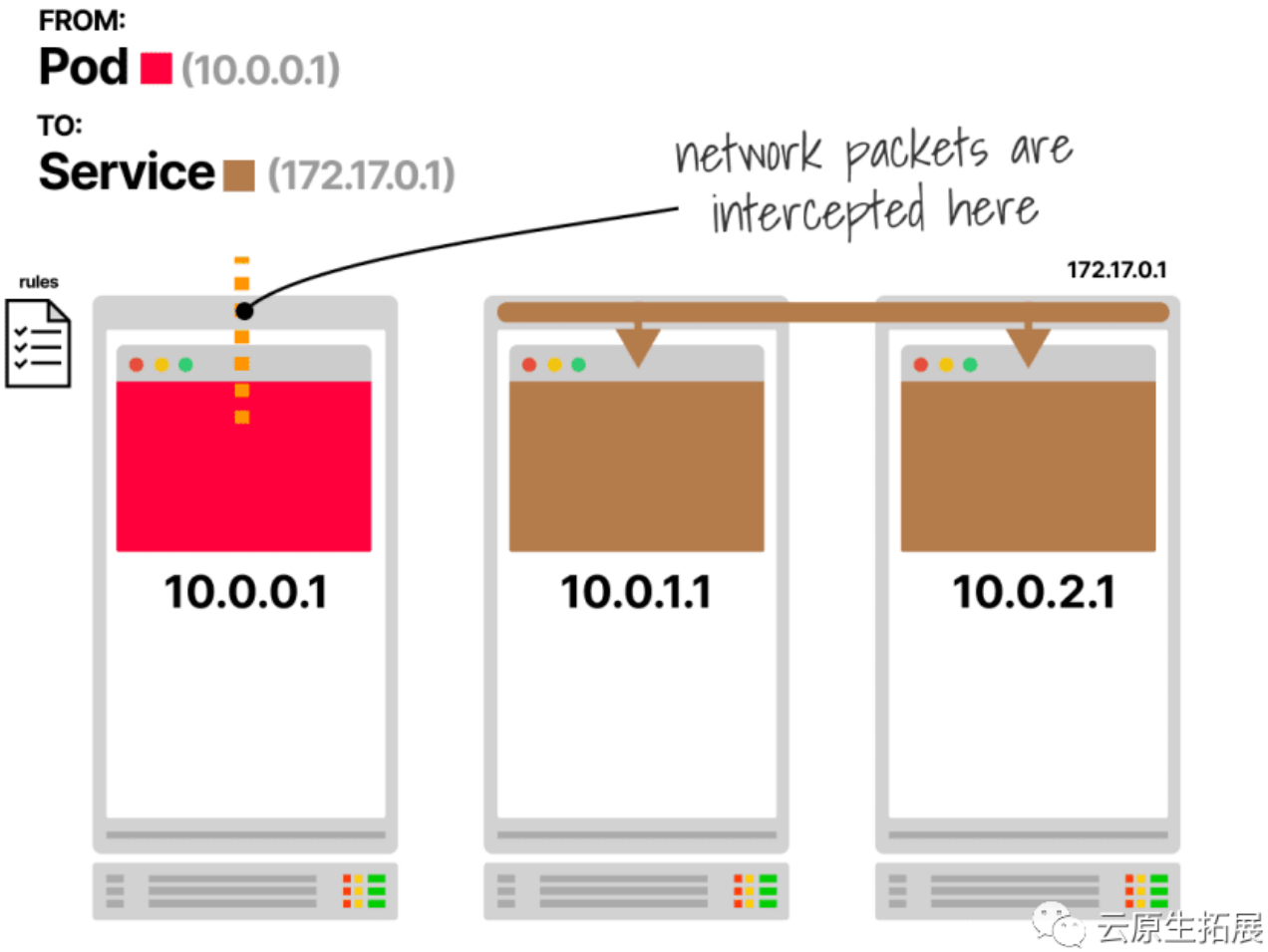
但是 Service 并不存在，它们的 IP 地址只是虚拟的。

流量如何到达其中一个 pod？

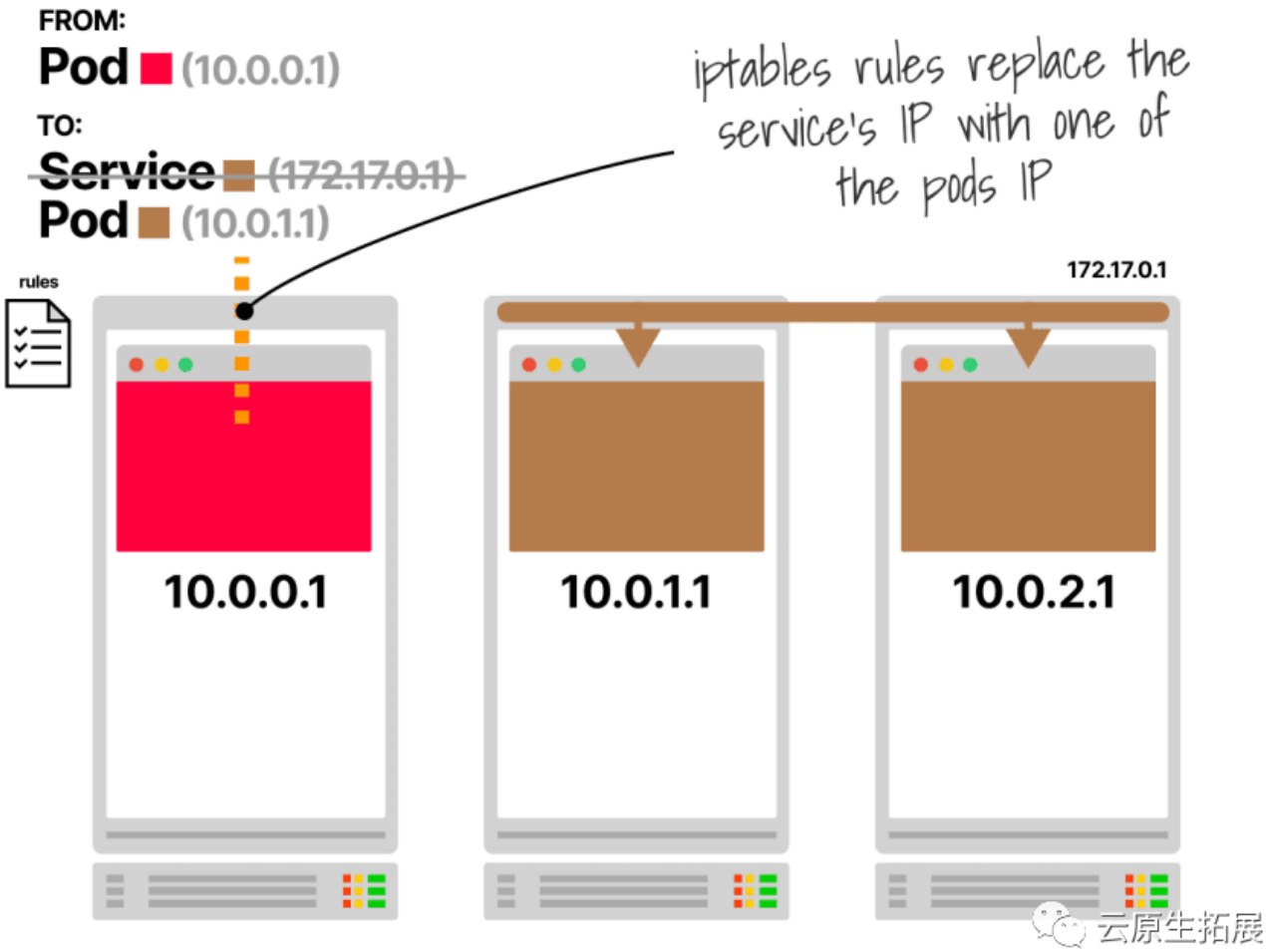


Kubernetes 使用了一个非常聪明的技巧。

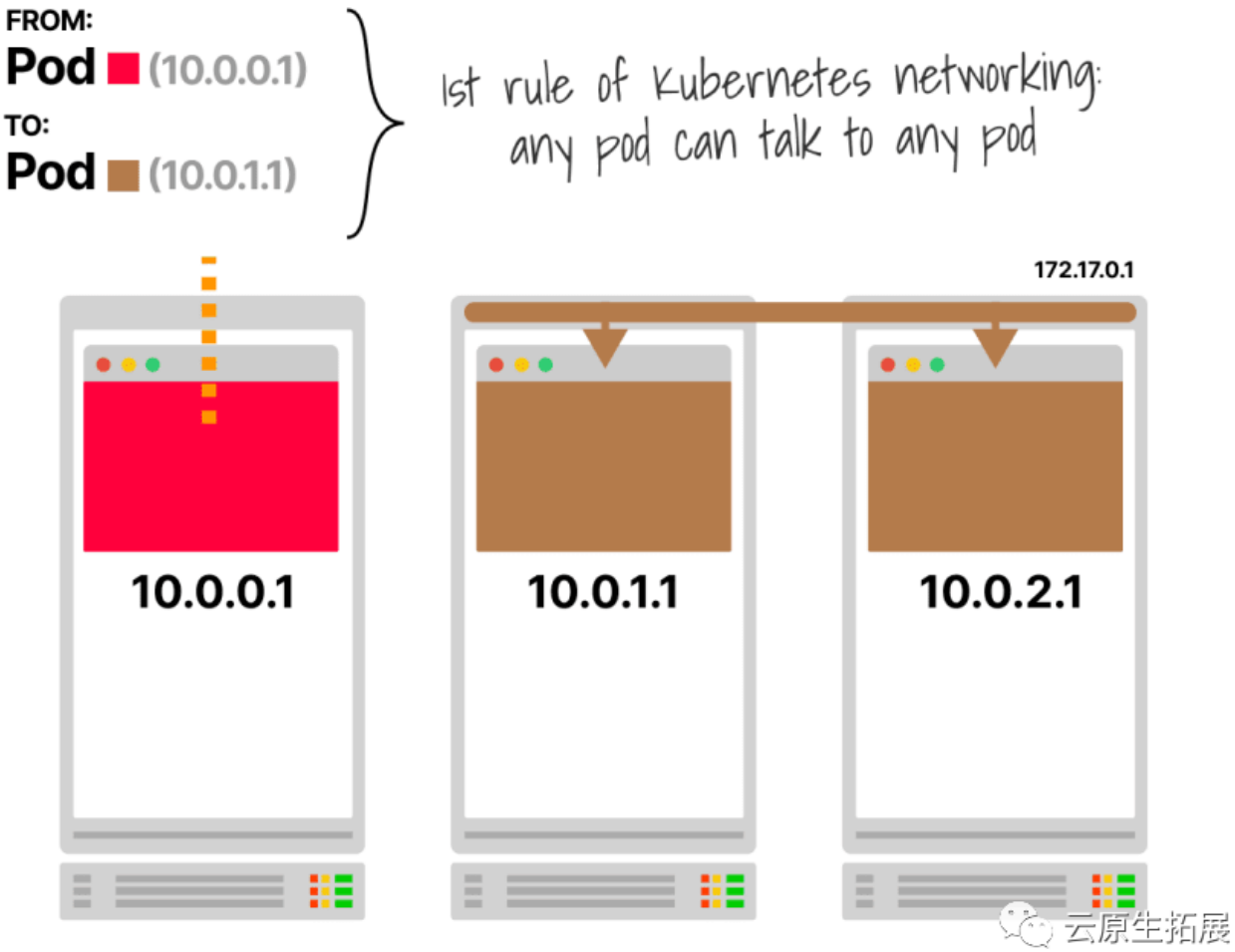
在请求退出节点之前，它被 **iptables 规则拦截** 。

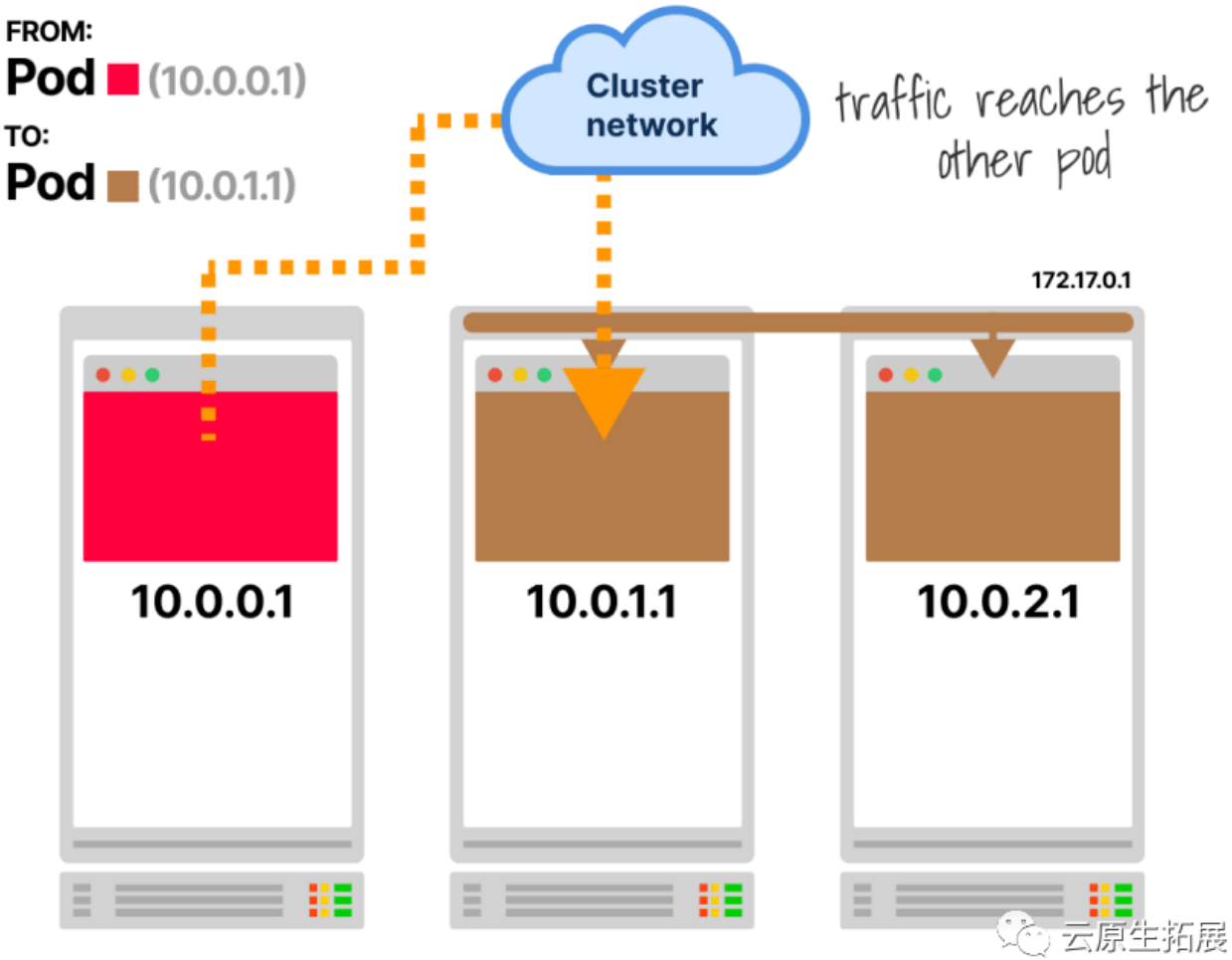


iptables 规则知道该 Service 不存在，并继续将 Service 的 IP 地址替换为属于该 Service 的 Pod 的 IP 地址之一。



目的地是一个 pod IP 地址，由于 Kubernetes 保证任何 pod 都可以与集群中的任何其他 pod 通信，因此流量可以流向棕色 pod。


FROM:
Pod (10.0.0.1)
TO:
Pod (10.0.1.1)
1st rule of Kubernetes networking:
any pod can talk to any pod
172.17.0.1
10.0.0.1
10.0.1.1
10.0.2.1

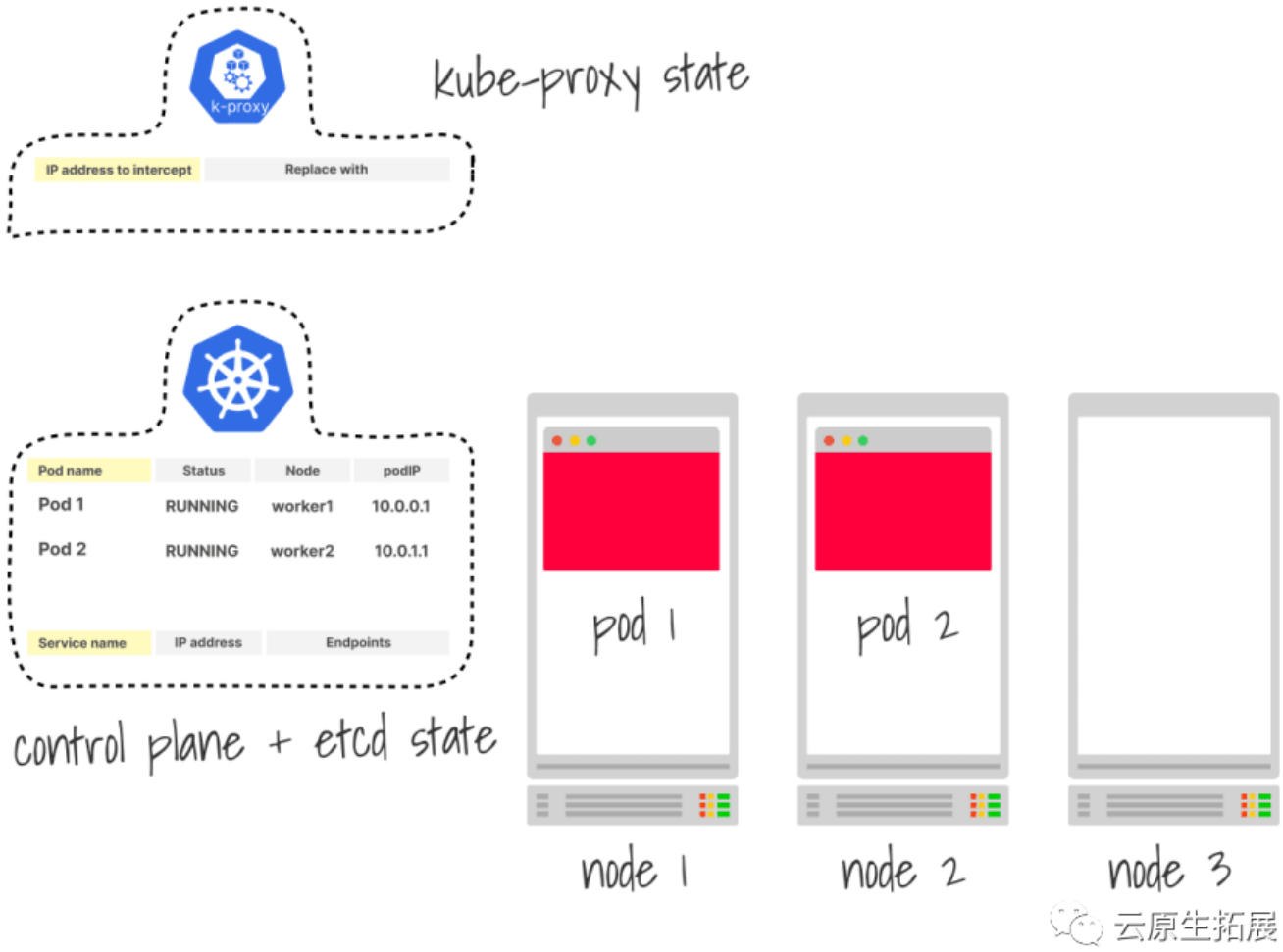

FROM:
Pod (10.0.0.1)
TO:
Pod (10.0.1.1)
Cluster network
traffic reaches the other pod
172.17.0.1
10.0.0.1
10.0.1.1
10.0.2.1

谁在配置这些 iptables 规则？

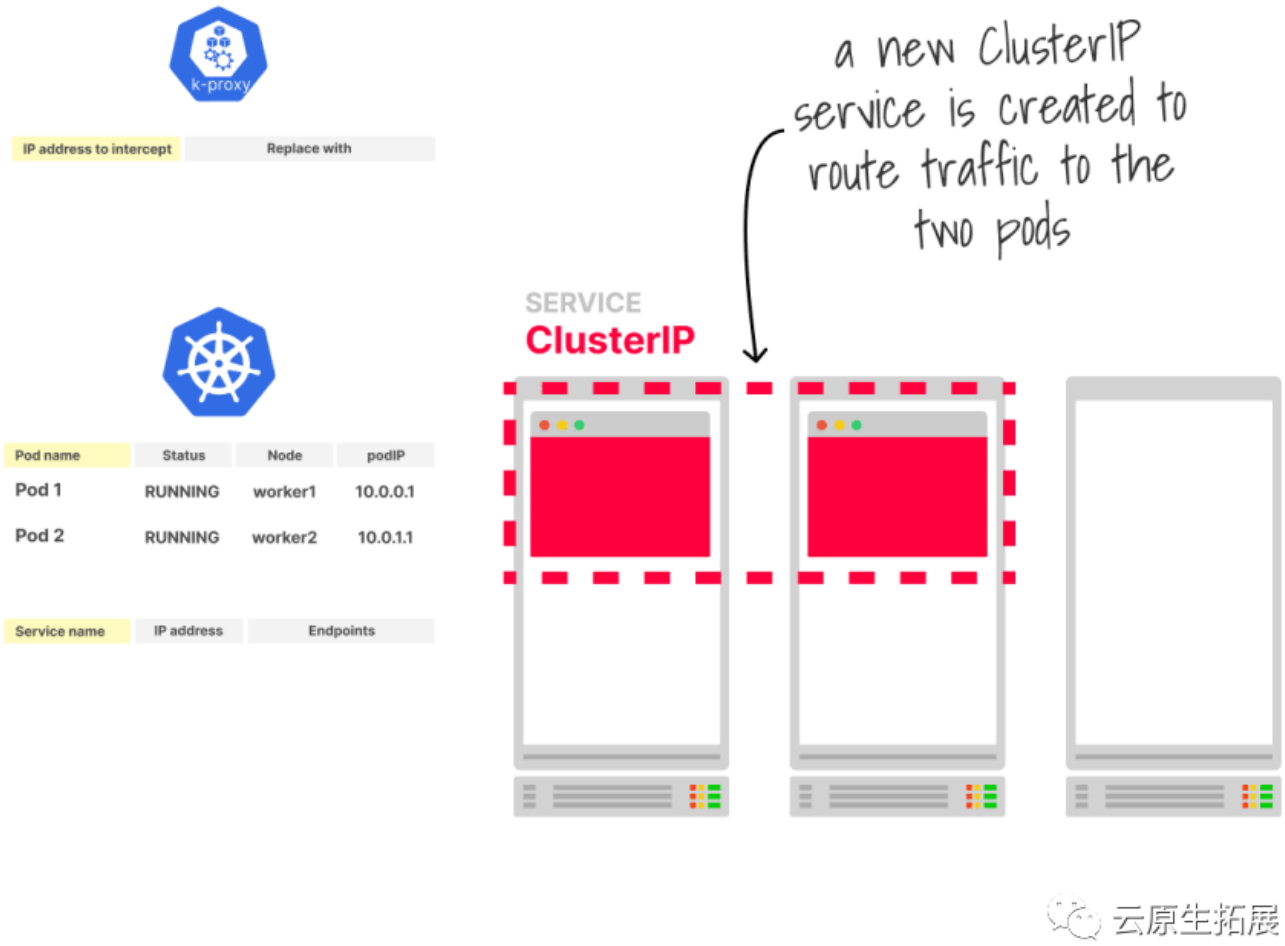
它是 kube-proxy 从控制平面收集端点 并将 Service IP 地址映射到 pod IP（它还对连接进行负载平衡）。

Kube-proxy 是一个 DaemonSet，它监听 Kubernetes API 的变化。

让我们来看看它是如何工作的。

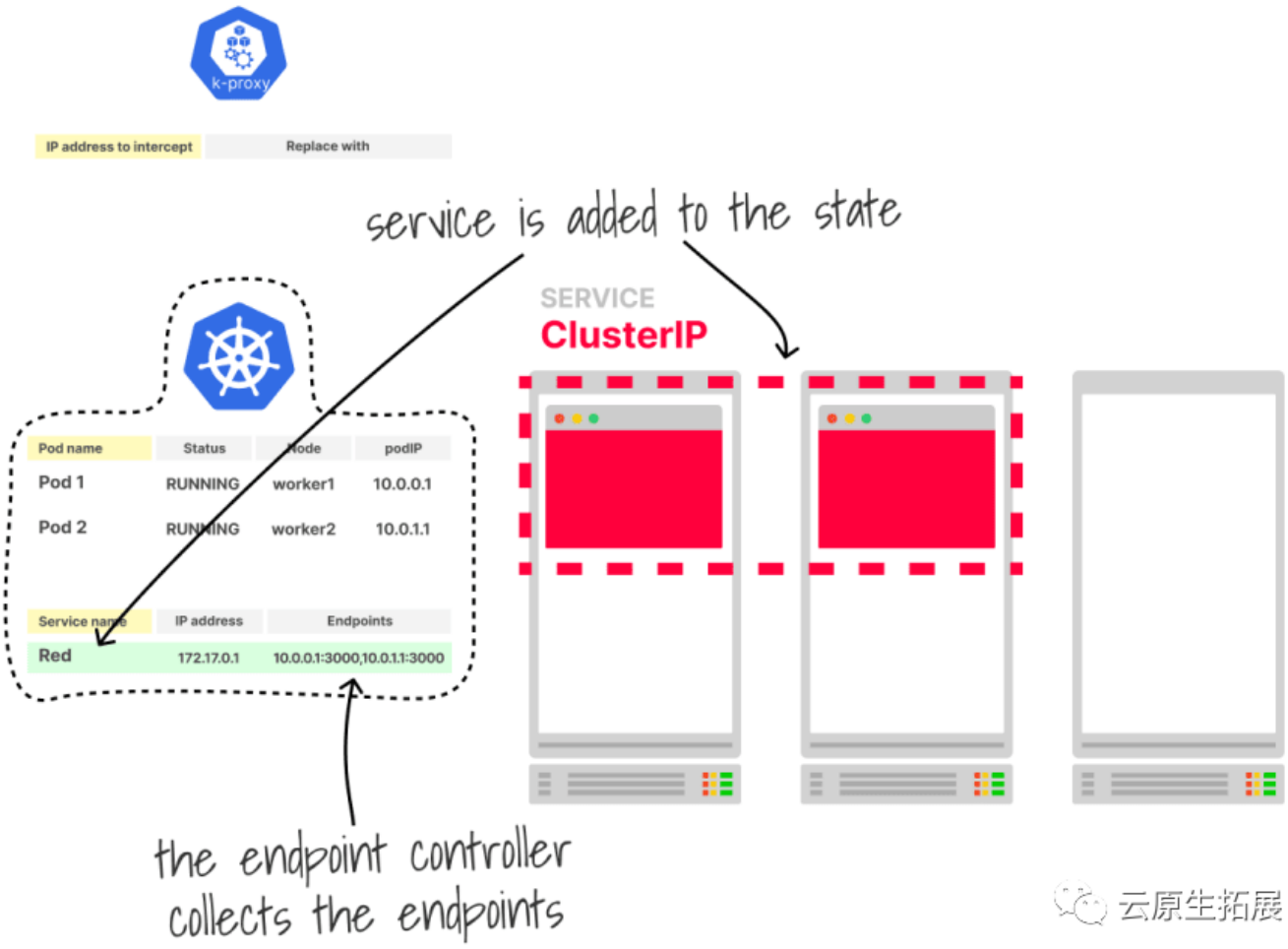


让我们观察创建 ClusterIP Service 时会发生什么。



在控制平面中分配一个固定的虚拟 IP 地址，并创建一个伴随端点对象。

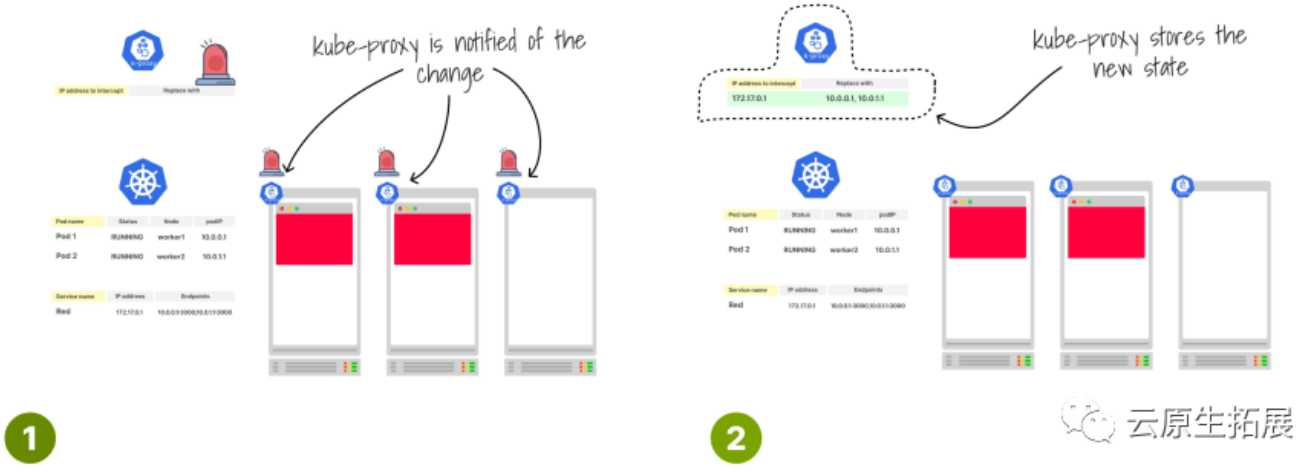
端点包含 IP 地址列表和应转发流量的端口。



Kube-proxy 订阅对控制平面的更改。

对于每个端点添加、删除或更新，它都会收到通知。

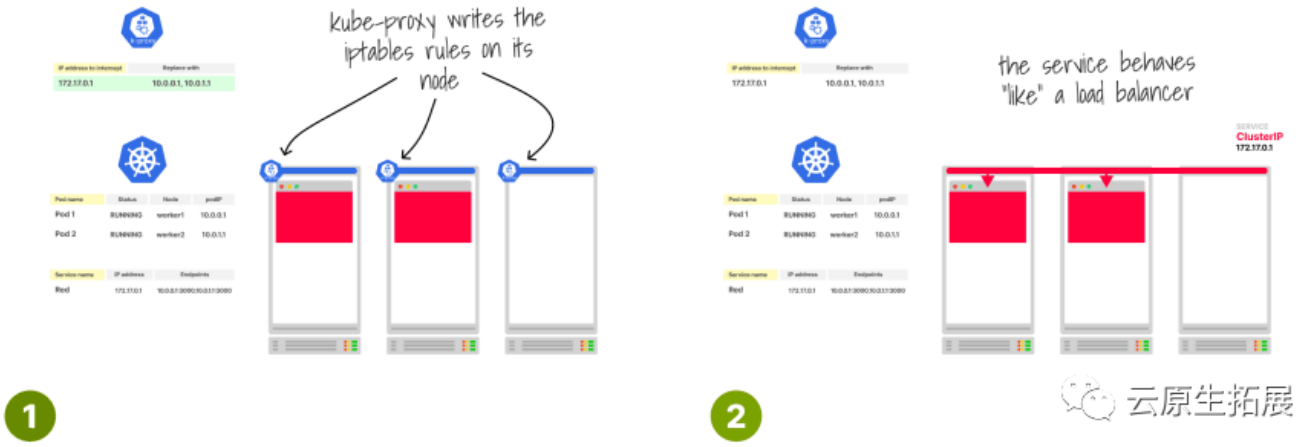
在这种情况下有一个新的Service（和端点对象）。



Kube-proxy 使用新的 **iptables** 规则列表更新其节点。

由于集群中的每个节点都有一个 kube-proxy，因此每个节点都将经历相同的过程。

最后，service “准备就绪”。



这解释了 Service 如何不存在以及 kube-proxy 如何在每个节点上设置负载平衡规则但没有回答为什么（有时）您无法 ping 通服务。

答案很简单： **iptables 中没有针对 ICMP 流量的规则** 。

所以 iptables 会跳过数据包。

但是由于 Service IP 是虚拟的（它不存在于 etcd 之外的任何地方），流量不会被拦截并且无处可去。

那么为什么它可以在我的集群上运行呢？

iptables 不是实现 ClusterIP Service 的唯一机制。

其他选项包括 IPVS 和 eBPF 等技术，它们的行为可能不同（取决于您使用的产品）。

这只是冰山一角。