

76Kubernetes 系列（六十九）在 Kubernetes 中创建 Pod 时发生了什么

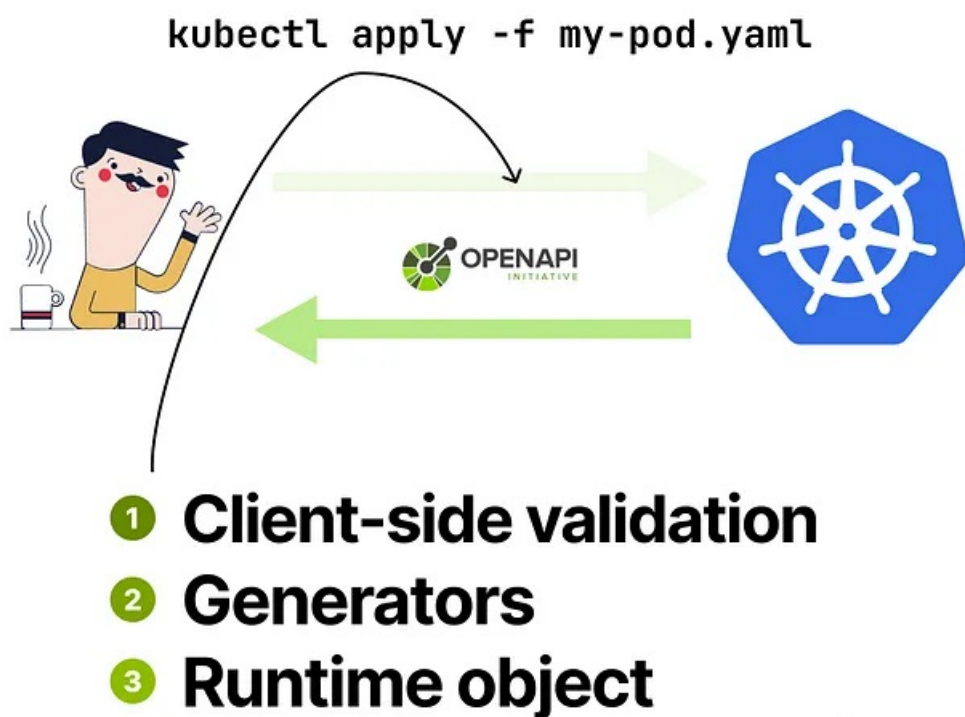
当你在 Kubernetes 中创建 Pod 时会发生什么？

一个令人惊讶的简单任务揭示了一个复杂的工作流程，该工作流涉及集群中的多个组件。

让我们从显而易见的开始：**kubectl** 将 **YAML** 定义发送到 **API Server**。

在此步骤中，**kubectl**：

- 使用 OpenAPI（Swagger）发现 API 端点。
- 协商资源版本。
- 验证 YAML。
- 发出请求。

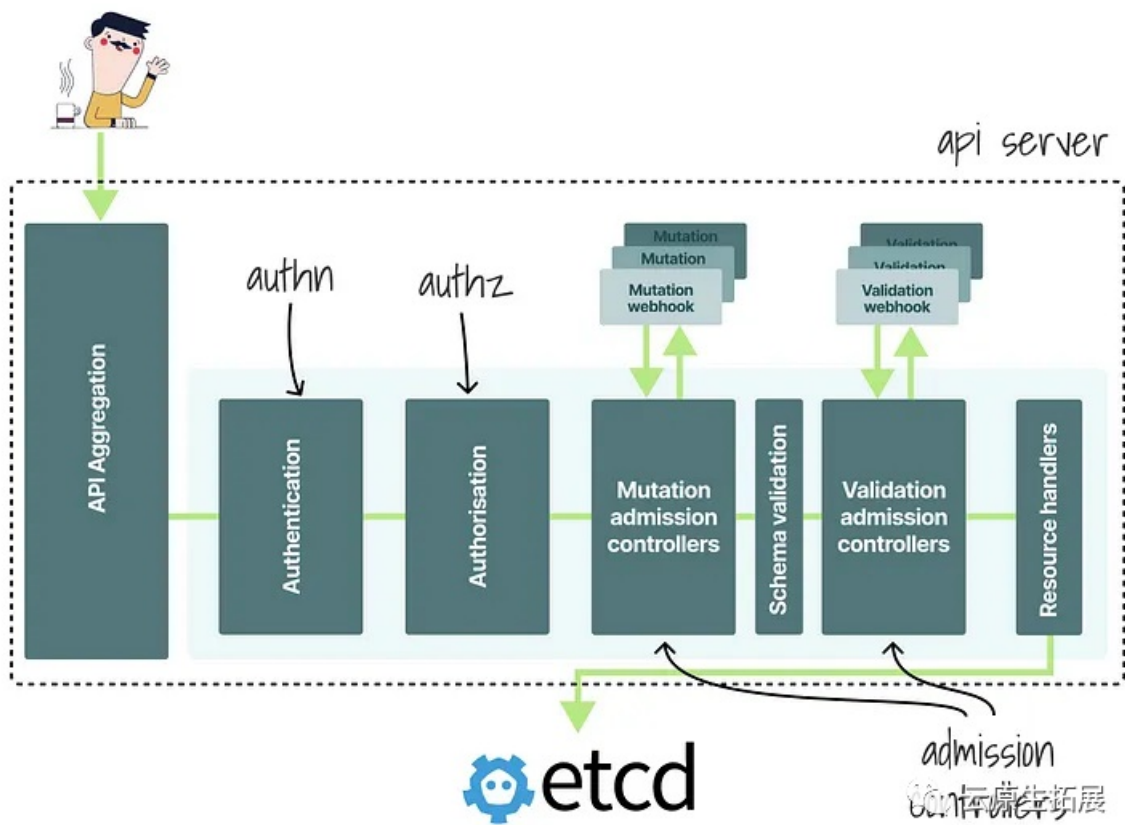


云原生拓展

当请求到达 API 时，它会经历以下过程：

- 身份验证和鉴权。
- 准入控制器。

当请求到达 API 时，它会经历以下过程：

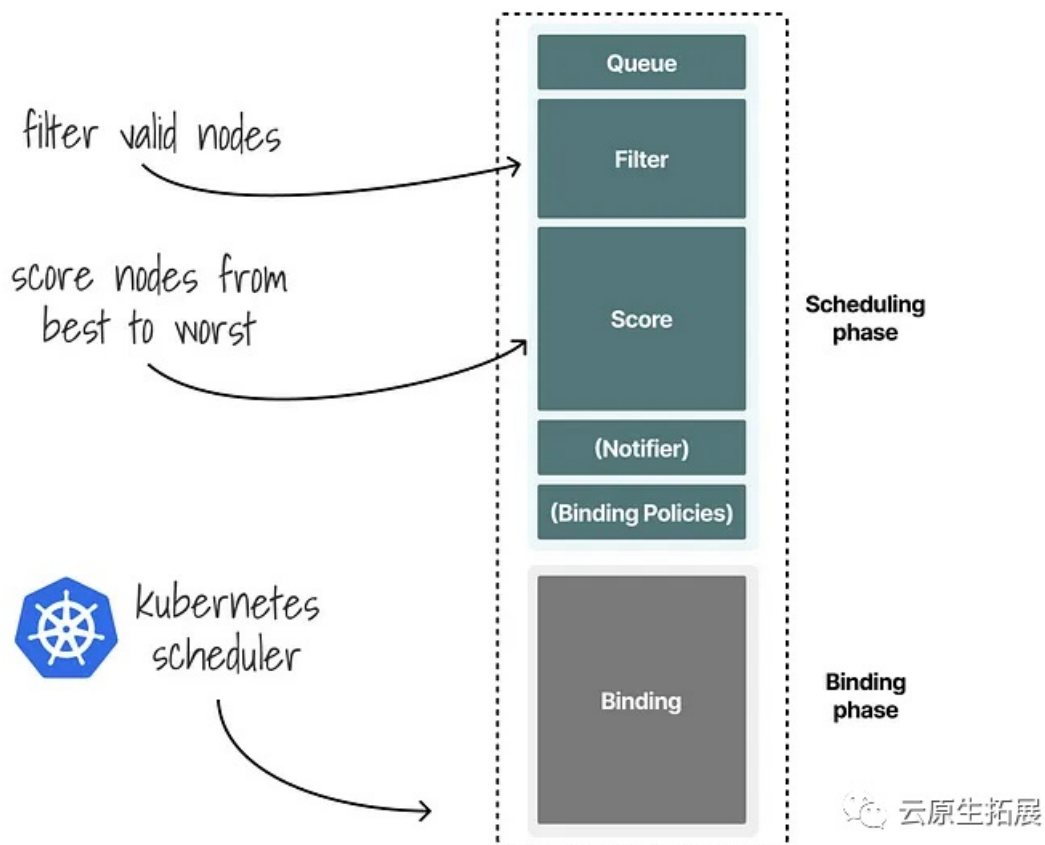


在此之后，Pod 将添加到调度程序队列中。

调度程序对节点进行过滤和评分，以找到最佳节点。

它最终将 Pod 绑定到节点。

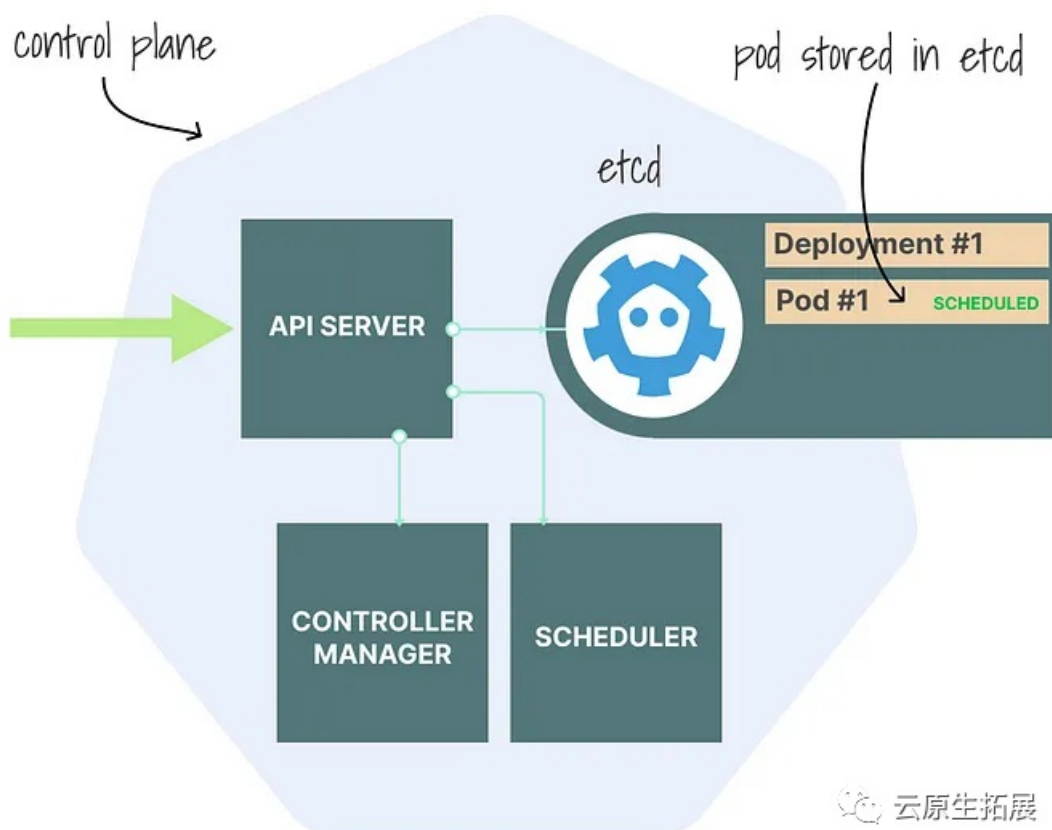
绑定写入到 etcd 中。



此时，pod 仅作为记录存在于 etcd 中。

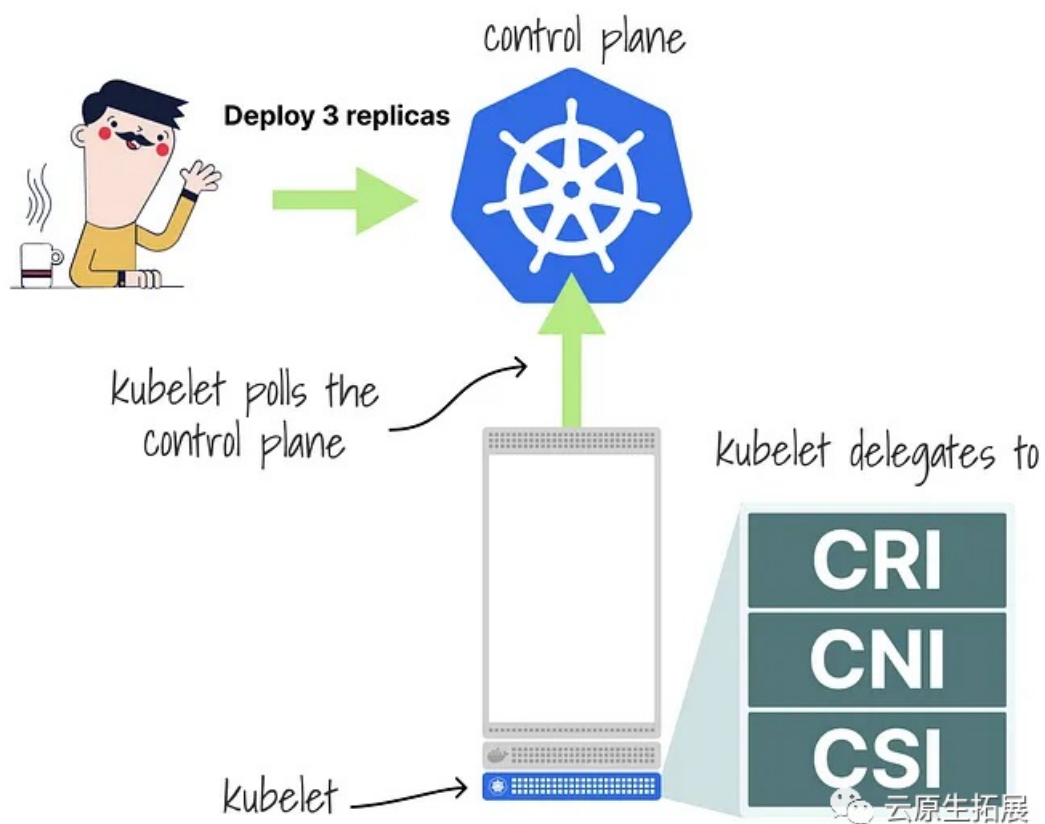
基础设施尚未创建任何容器。

接下来就是等到 kubelet 来接管了。



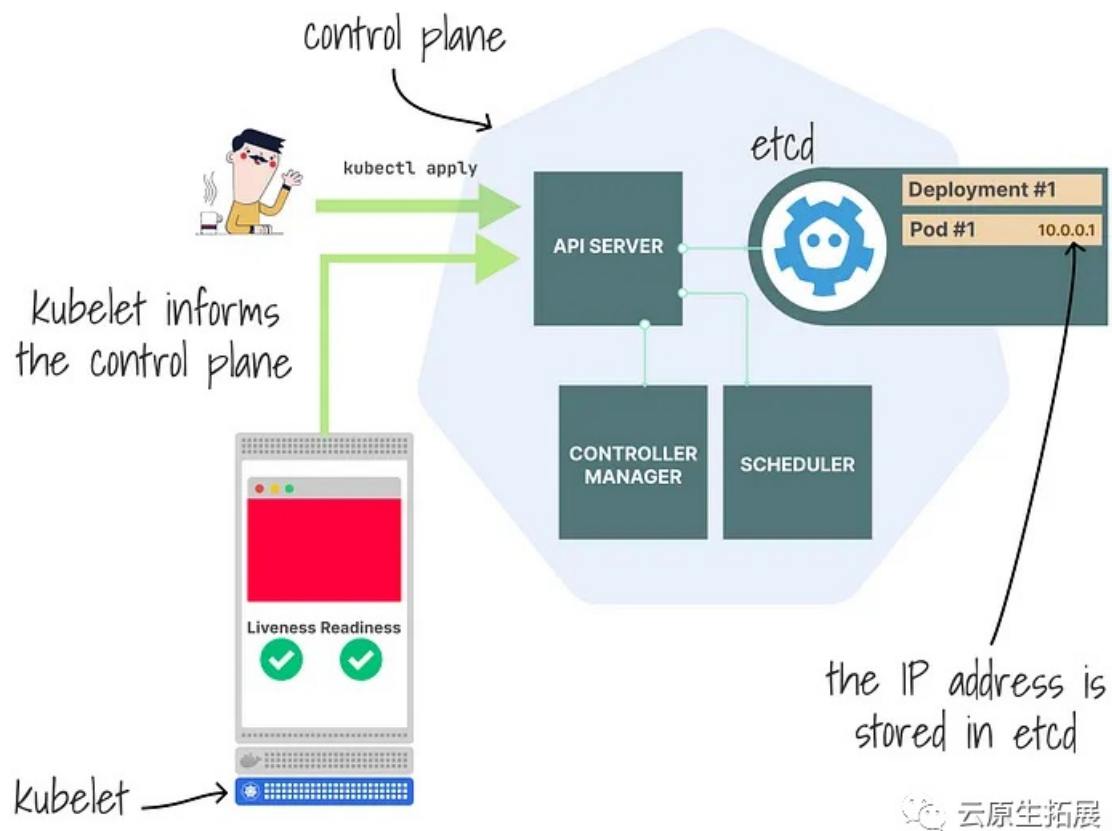
kubelet 拉取 Pod 定义并继续委托：

1. 创建到 CNI 的网络（例如 Cilium）。
2. 向 CRI 创建容器（例如 containerd）。
3. 向 CSI 创建存储（例如 OpenEBS）。



除此之外，Kubelet 将执行 Pod 的探测，并在 Pod 运行时将其 IP 地址报告给控制平面。

该 IP 和容器的端口作为端点存储在 etcd 中。



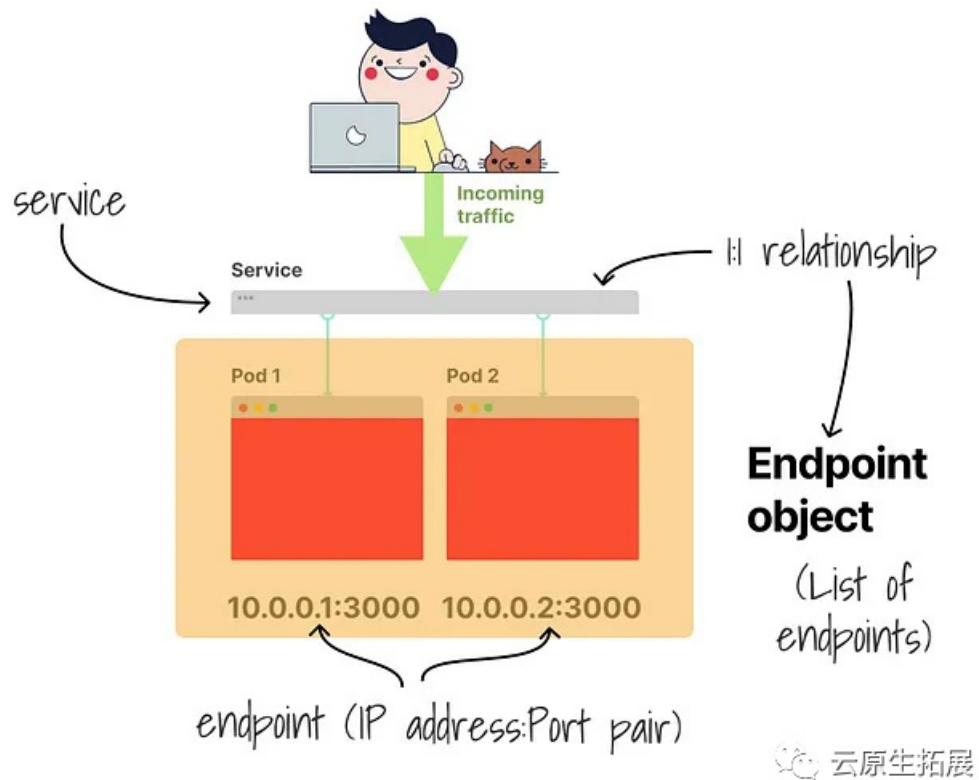
等等。。。端点是什么？

在 Kubernetes 中：

- endpoints 是类似 `10.0.0.2:3000(IP:port)` 对。
- Endpoint 对象是 endpoints 的集合（IP:port 对的列表）

对于集群中的每个服务，Kubernetes 都会创建一个包含 endpoints 的 Endpoint 对象。

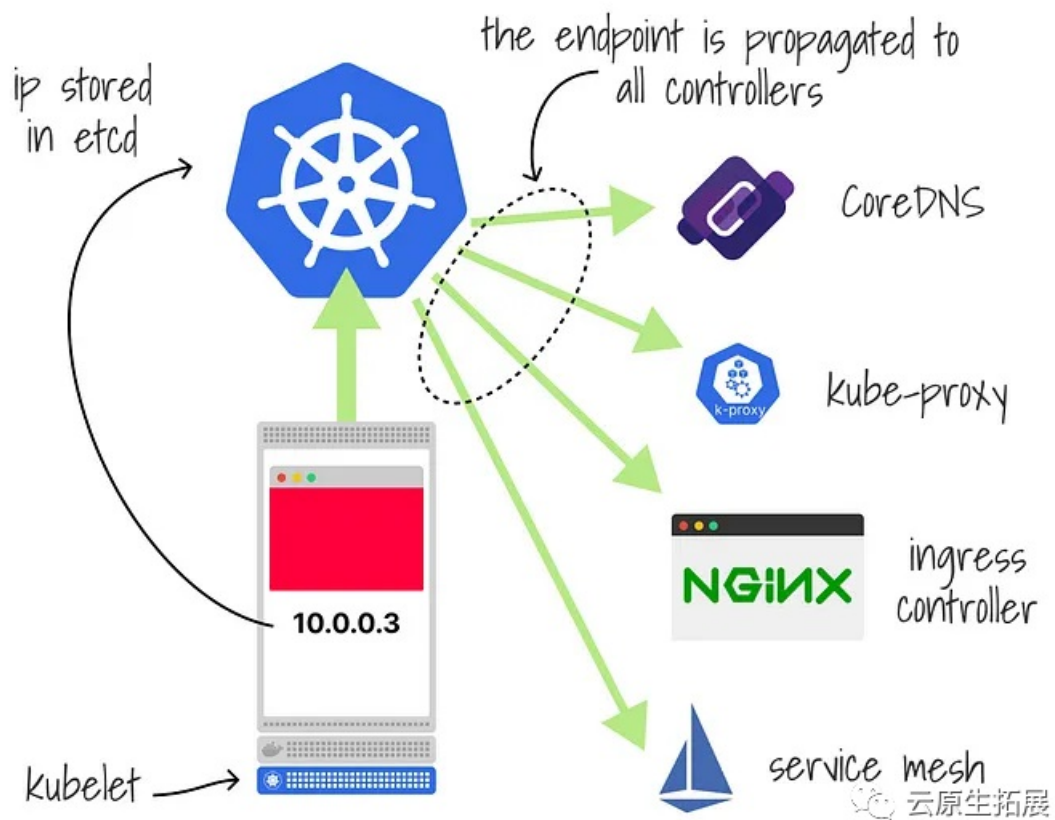
令人困惑，不是吗？



端点（IP：端口）由以下相关目标使用：

- kube-proxy 用它来设置 iptables 规则。
- 用于 CoreDNS 更新 DNS 条目。
- 用于 Ingress 控制器设置下游。
- 服务网格。
- 以及更多的 operators。

添加端点后，将立即通知相关组件。



当端点（IP：端口）传播完成时，你终于可以开始使用 Pod 了！

删除 Pod 时会发生什么情况？

过程完全相反。

正确的顺序是：

1. 应用停止接受连接。
2. 控制器（kube-proxy、Ingress 等）删除端点。
3. 应用程序耗尽现有连接。
4. 关闭应用。

