

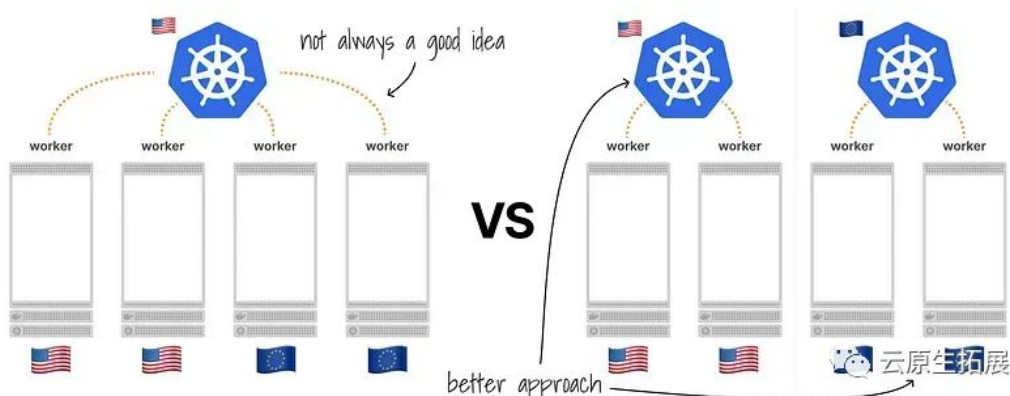
68Kubernetes 系列（六十二）将 Kubernetes 扩展到多个集群和区域

在本教程中，您将学习如何在不同地区创建、连接和操作三个 Kubernetes 集群：北美、欧洲和东南亚。

Kubernetes 的一个有趣挑战是跨多个区域部署工作负载。

虽然从技术上讲，您可以拥有一个包含位于不同区域的多个节点的集群，但由于额外的延迟，这通常被认为是您应该避免的事情。

另一种流行的替代方法是为每个区域部署一个集群，并找到一种方法来编排它们。



但在讨论解决方案之前，让我们先看看多集群和多云设置的挑战。

当你编排多个集群时，你不得不面对以下问题：

- 您如何决定分配工作负载？
- 网络如何跨区域工作？
- 我应该如何处理有状态的应用程序/数据？

CHALLENGES OF MULTICLUSTER

- 1 Scheduling**
how should the pod be distributed?
- 2 Networking (north-south)**
how do I route the traffic to the right cluster?
- 3 Networking (east-west)**
should pod in one cluster be able to reach pods in others?
- 4 Data & stateful apps**
if a pod is rescheduled in a different cluster, what should you do with the data?

云原生拓展

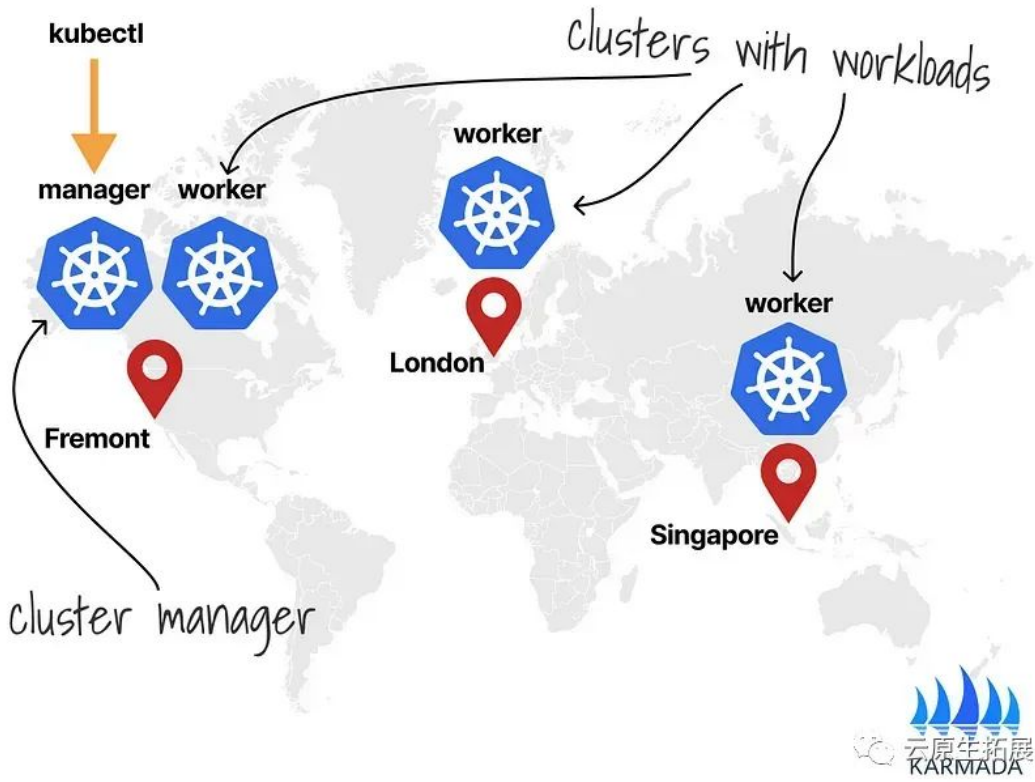
让我们试着回答其中的一些问题。

为了解决第一个问题（调度工作负载），我使用了 Karmada(<https://karmada.io/>)。

借助 **Karmada**，您可以使用 **kubectl** 创建部署并使用策略将它们分布在多个集群中。

Karmada 负责将它们传播到正确的集群。

该项目（在精神上）类似于 kubefed（<https://github.com/kubernetes-sigs/kubefed>）。



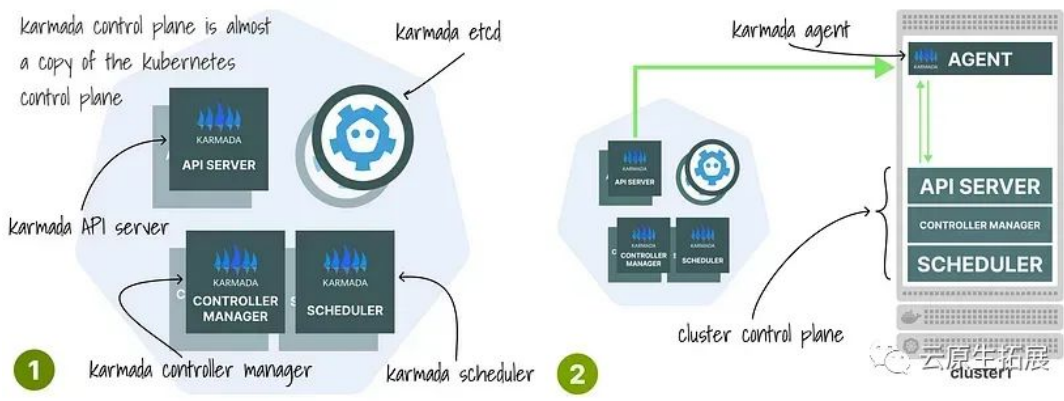
Karmada 使用 **Kubernetes** 集群作为管理器，并创建第二个支持多集群的控制平面。

这特别方便，因为 **kubectl** “可以正常工作”并且现在支持多集群。

换句话说，你可以继续使用**kubectl**，但是所有的命令都可以跨集群申请资源和聚合数据。

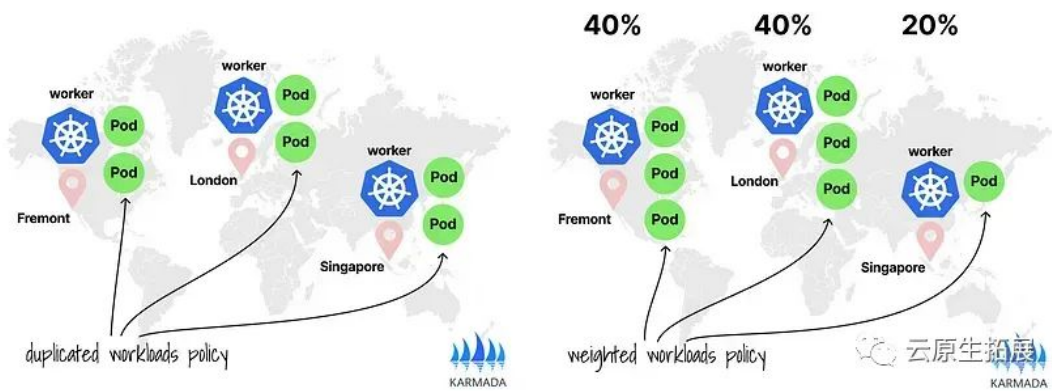
每个集群都有一个向集群的 **API** 服务器发出命令的代理。

Karmada 控制器管理器使用这些代理来同步和分发命令。



Karmada 使用策略来决定如何分配您的工作负载。（<https://karmada.io/docs/userguide/scheduling/resource-propagating>）

您可以制定策略，让部署在各个区域平均分布。或者你可以将你的 **pod** 放在一个区域中。



Karmada 本质上是一个多集群协调器，但不提供任何连接集群网络的机制。

路由到某个区域的流量将始终到达该区域的 pod。



但是您可以使用像 Istio 这样的服务网格来创建一个跨越多个集群的网络。

Istio 可以发现其他集群中的其他实例，并将流量转发到其他集群。

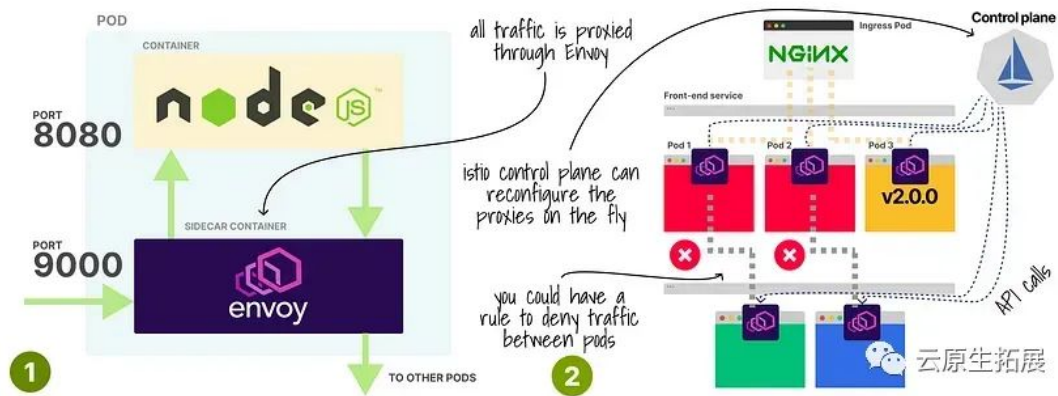


但是流量路由是如何工作的呢？

对于集群中的每个应用程序，Istio 都会注入一个边车代理。

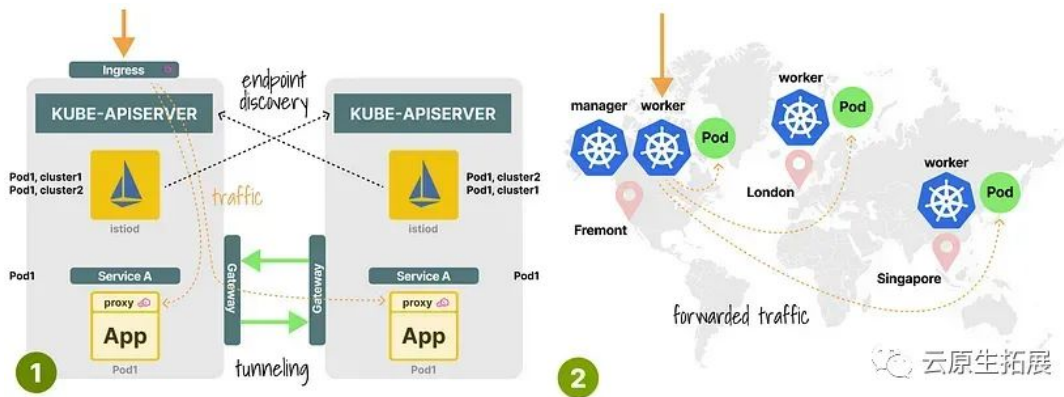
进出应用程序的所有流量都通过代理。

Istio 控制平面可以动态配置代理并应用路由策略。



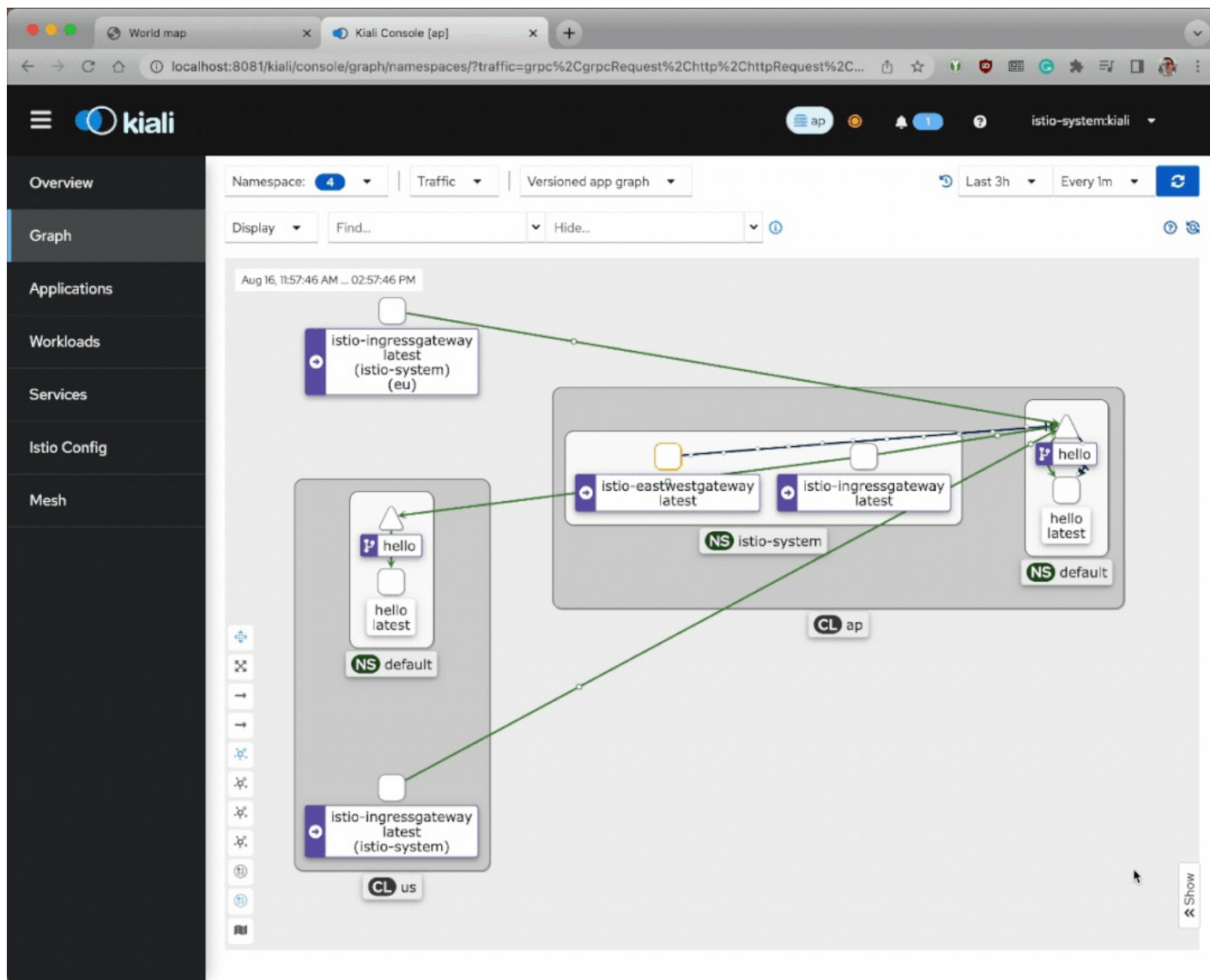
在多集群设置中，Istio 实例共享端点。

当发出请求时，流量被代理边车拦截并转发到所有集群中所有端点中的一个端点。



由于 Istio 的流量路由规则可以让您轻松控制服务之间的流量和 API 调用的流动(<https://istio.io/latest/docs/concepts/traffic-management/>)，因此即使在每个区域部署 pod，您也可以让流量到达单个区域。

或者您可以创建规则将流量从一个区域转移到另一个区域。



如果你想看到它的实际效果，你可以在这里观看我的演示(<https://event.on24.com/wcc/r/3919843/B94BC548B8902DD70F5EB512A308BB41?partnerref=learnk8s>)。