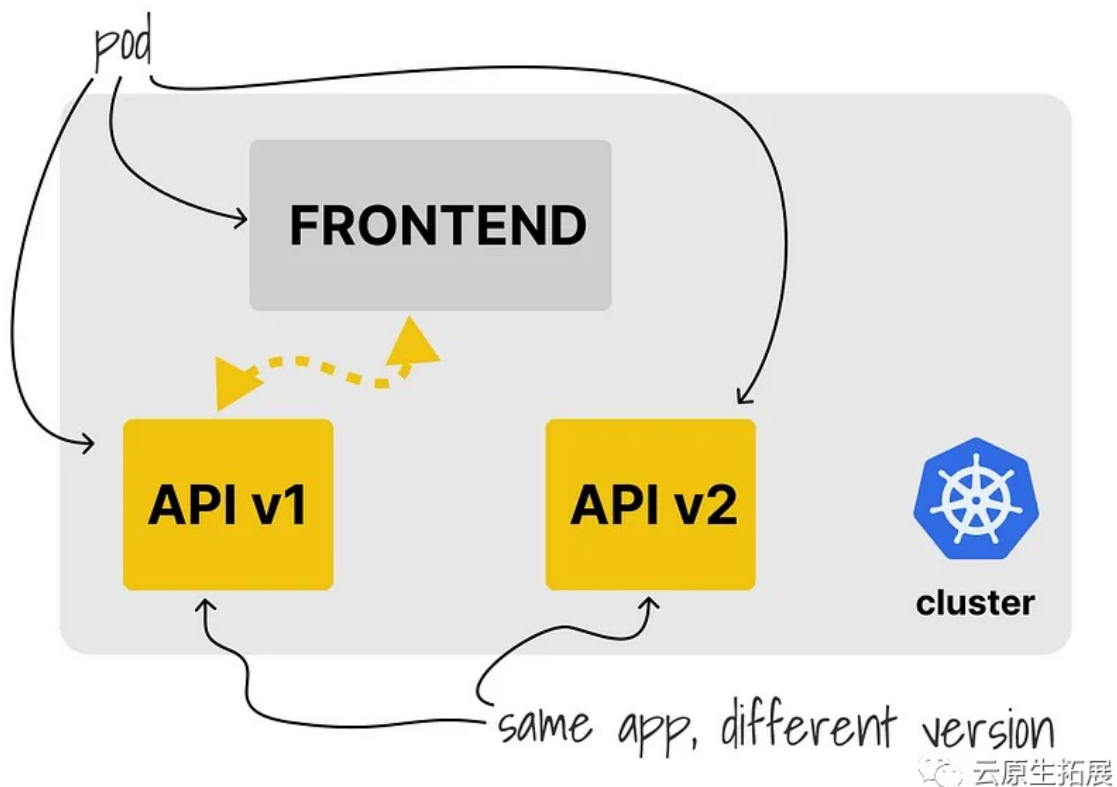


您可以使用 Istio、Kiali 和 Gateway API 的金丝雀版本向 Kubernetes 中的部分用户推出应用程序。

让我们从一个例子开始。

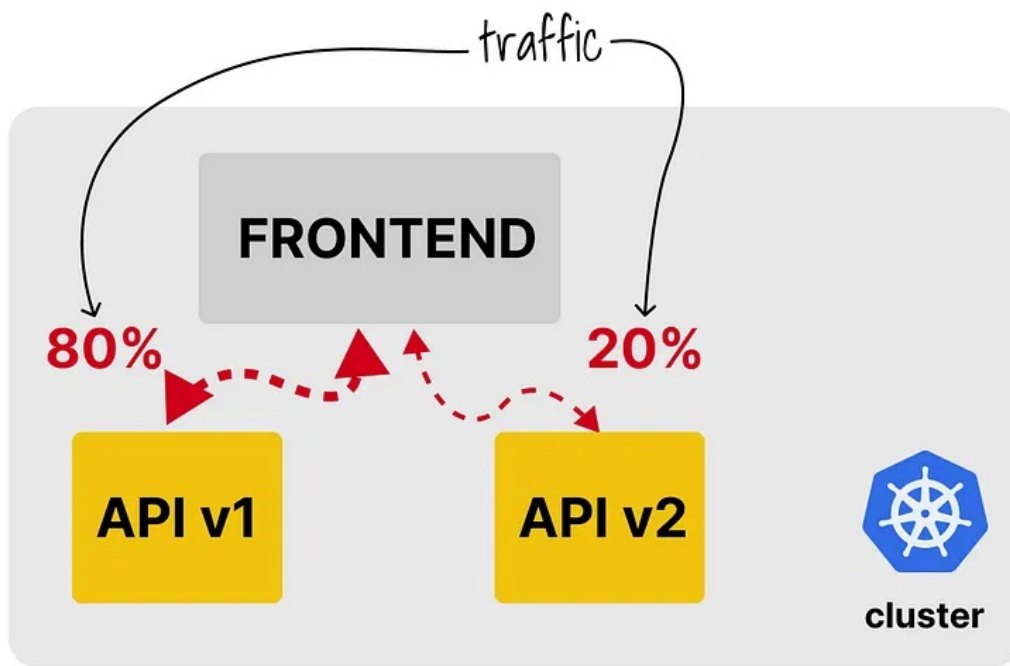
当前群集有三个应用：

1. 在版本 v1 公开 API 的后端。
2. 版本 v2 上的另一个应用。
3. 使用 API 的前端组件。



理想情况下，前端应消耗来自 v1 的 80% 的请求，仅消耗来自 v2 的 20% 的请求。

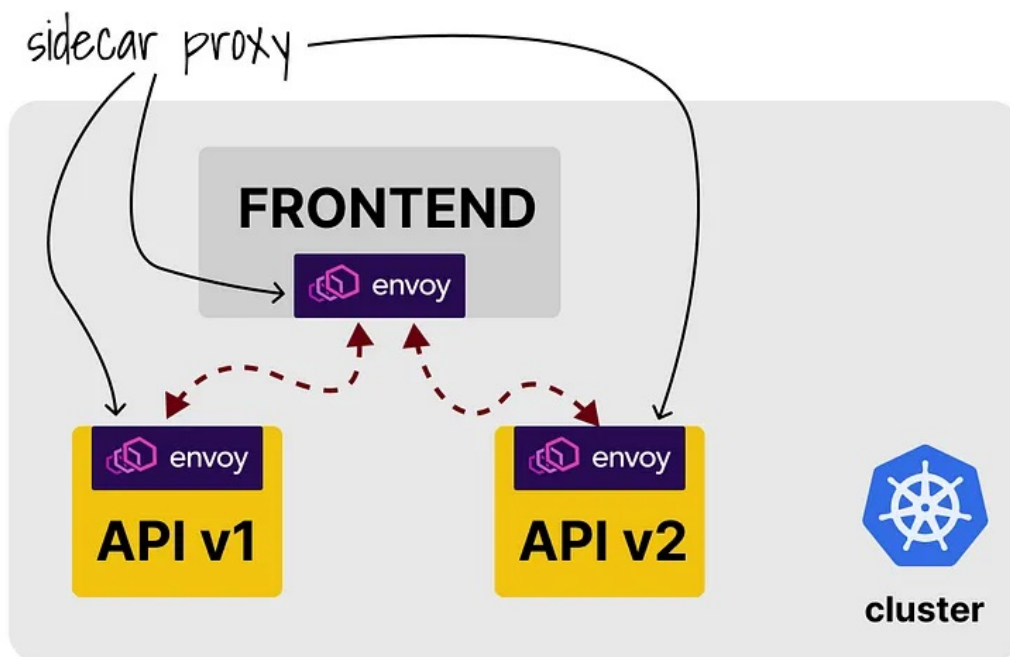
为此，您可以使用服务网格。



云原生拓展

安装服务网格后，群集中的每个 Pod 都会获得一个额外的容器。

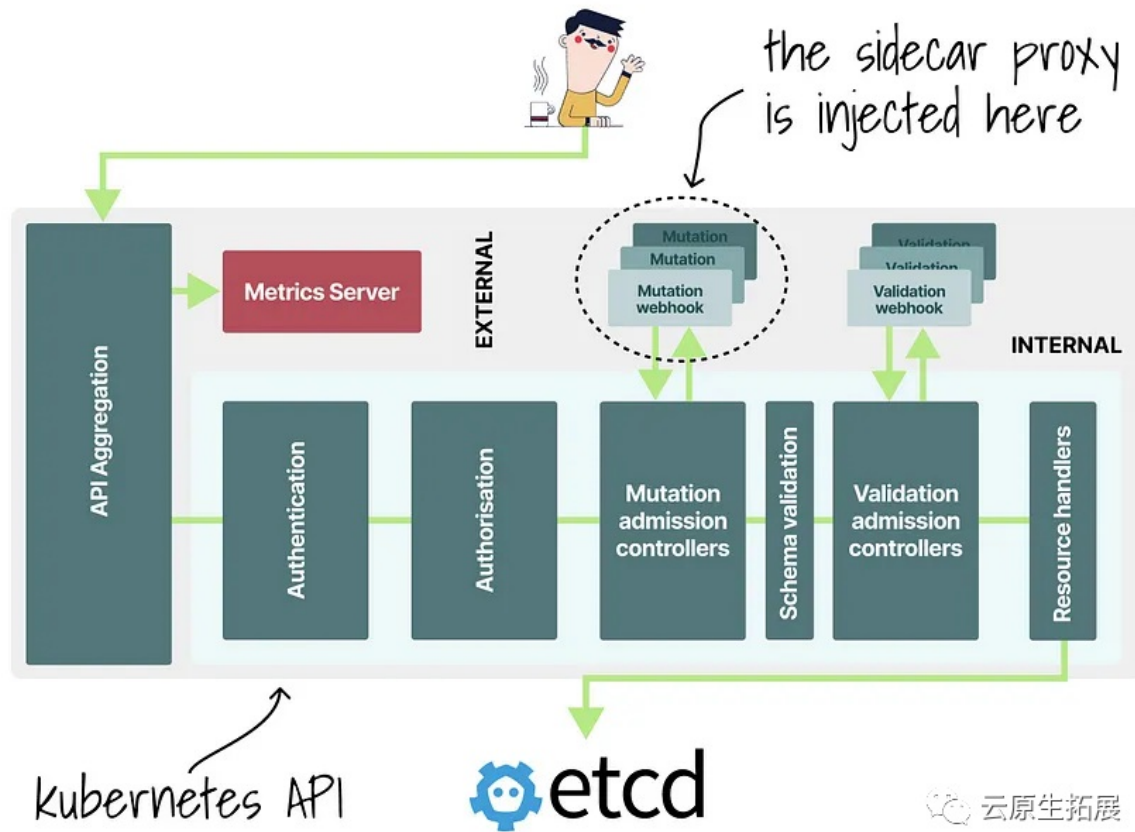
该容器代理所有传出和传入请求。



云原生拓展

代理是使用变异网络钩子（mutating webhook）自动注入的。

在 pod 存储在 etcd 中之前，会修改 YAML 定义，并注入代理。



服务网格非常有用，因为您可以：

- 监控指标。
- 跟踪组件之间的依赖关系。
- 决定流量拆分。

## WHY YOU MIGHT WANT TO USE A SERVICE MESH



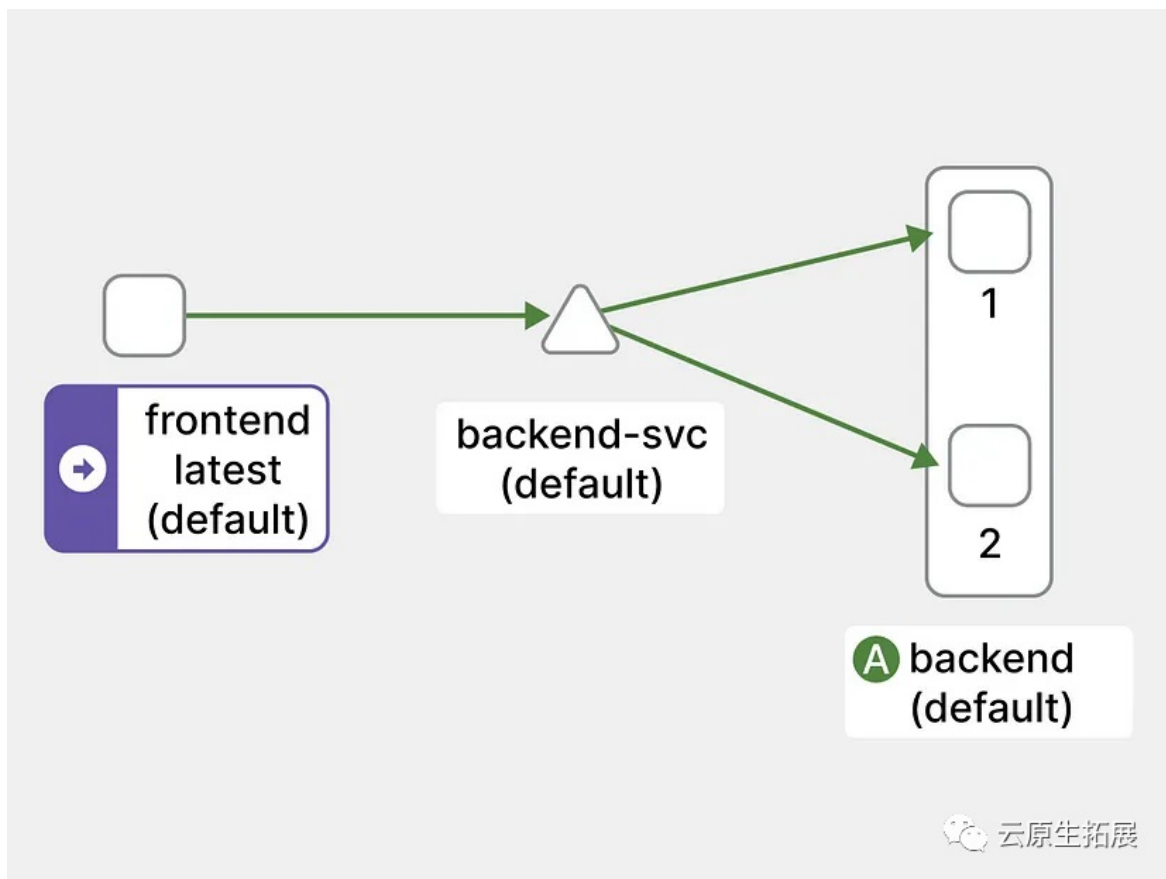
云原生拓展

我生成了一些流量来测试它并使用Kiali来跟踪它。

它自动映射了所有组件和交通方向。

所有这些都没有任何来自我这边的提示！

不过金丝雀发布呢？



可以使用服务网格来微调每个应用消耗的流量。

为了测试它，我在两个后端之间创建了一个 80-20 的拆分。

在这个例子中，我使用的是HTTPRoute：

```

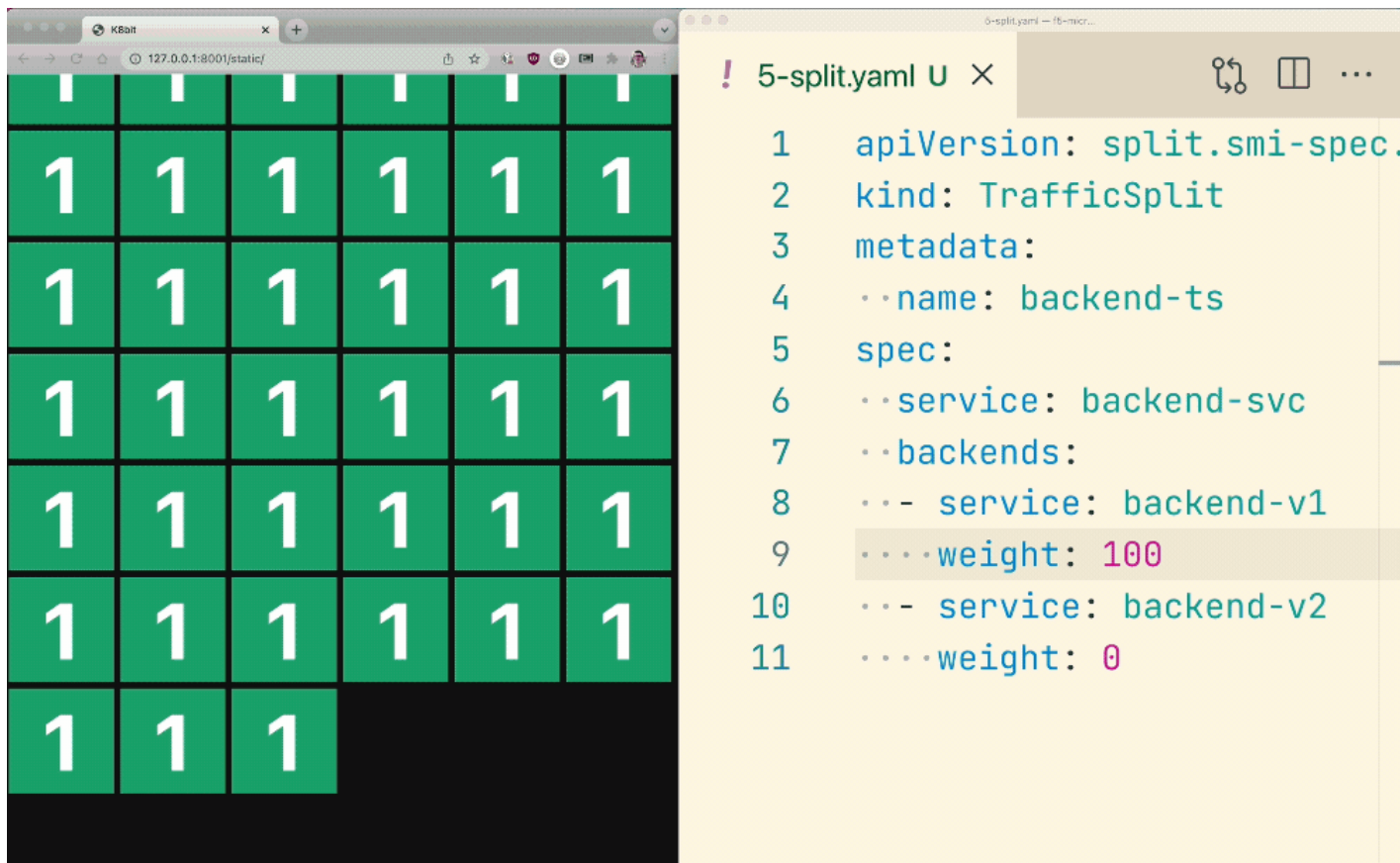
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: split
spec:
  parentRefs:
    - kind: Service
      name: backend-svc
      port: 80
  rules:
    - backendRefs:
        - name: backend-v1
          port: 80
          weight: 80
        - name: backend-v2
          port: 80
          weight: 20

```

Gateway API CRD

traffic split 80-20

HTTPRoute 是网关 API 的对象部分，可让您逐渐增加和减少流量，并可用于从 80-20 拆分转换为 0-100。



服务网格还可以：

1. 帮你推出影子版本。
2. 加密容器流量。
3. 镜像集群之间的流量。
4. 检查并重写流量。
5. 执行策略。
6. 注入故障以测试弹性。