

Apache APISIX Ingress 设计和实践

apisix.apache.org

张晋涛

个人介绍



- 张晋涛
- Apache APISIX committer
- Kubernetes Ingress Nginx reviewer
- containerd/Docker/Helm/Kubernetes contributor
- 『K8s 生态周报』 维护者
- <https://github.com/tao12345666333>



Agenda

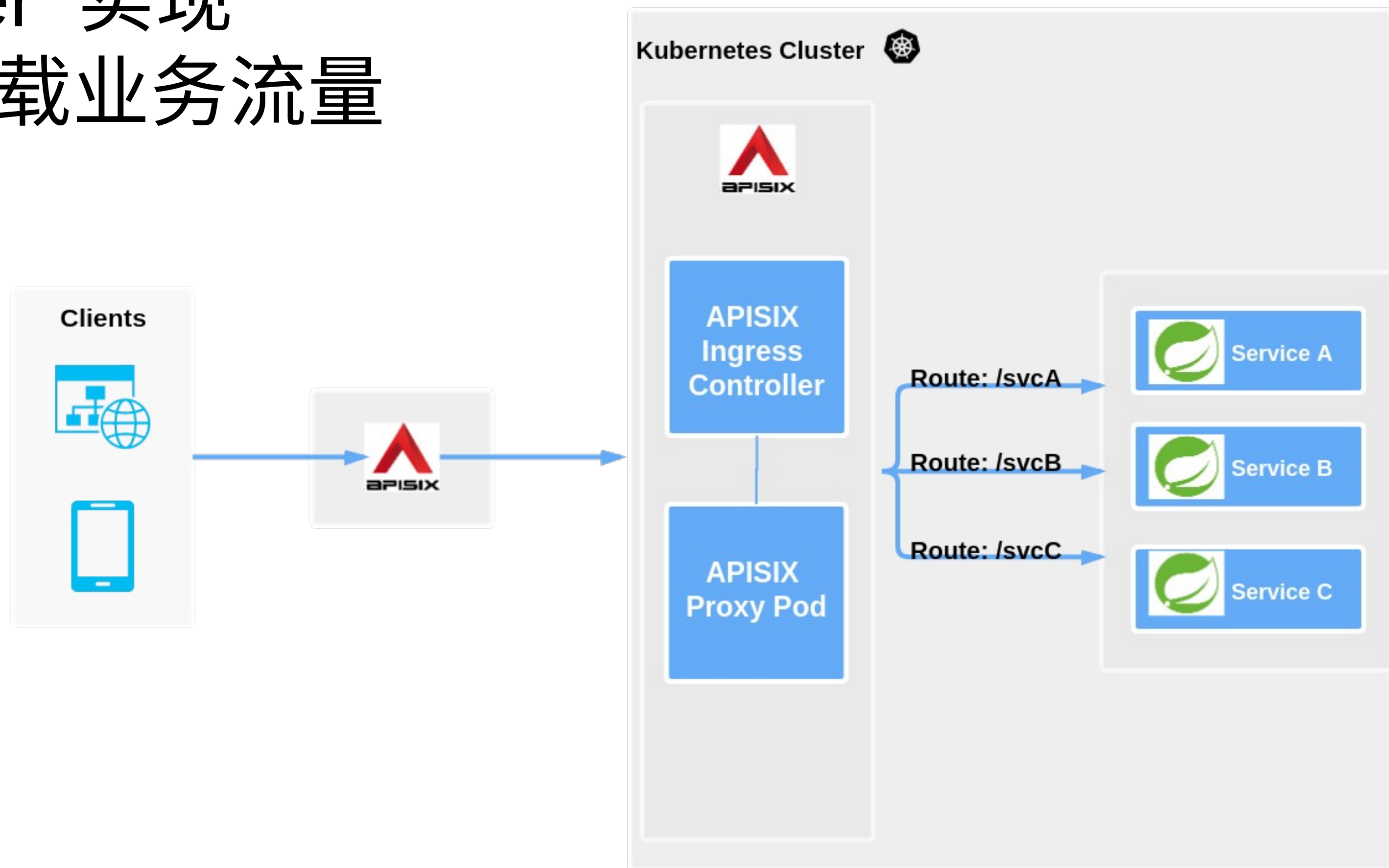


- Apache APISIX Ingress 概览
- Apache APISIX Ingress 的设计
- Apache APISIX Ingress 的上手实践
- Apache APISIX Ingress 后续规划

一、Apache APISIX Ingress 概览

APISIX Ingress 是什么

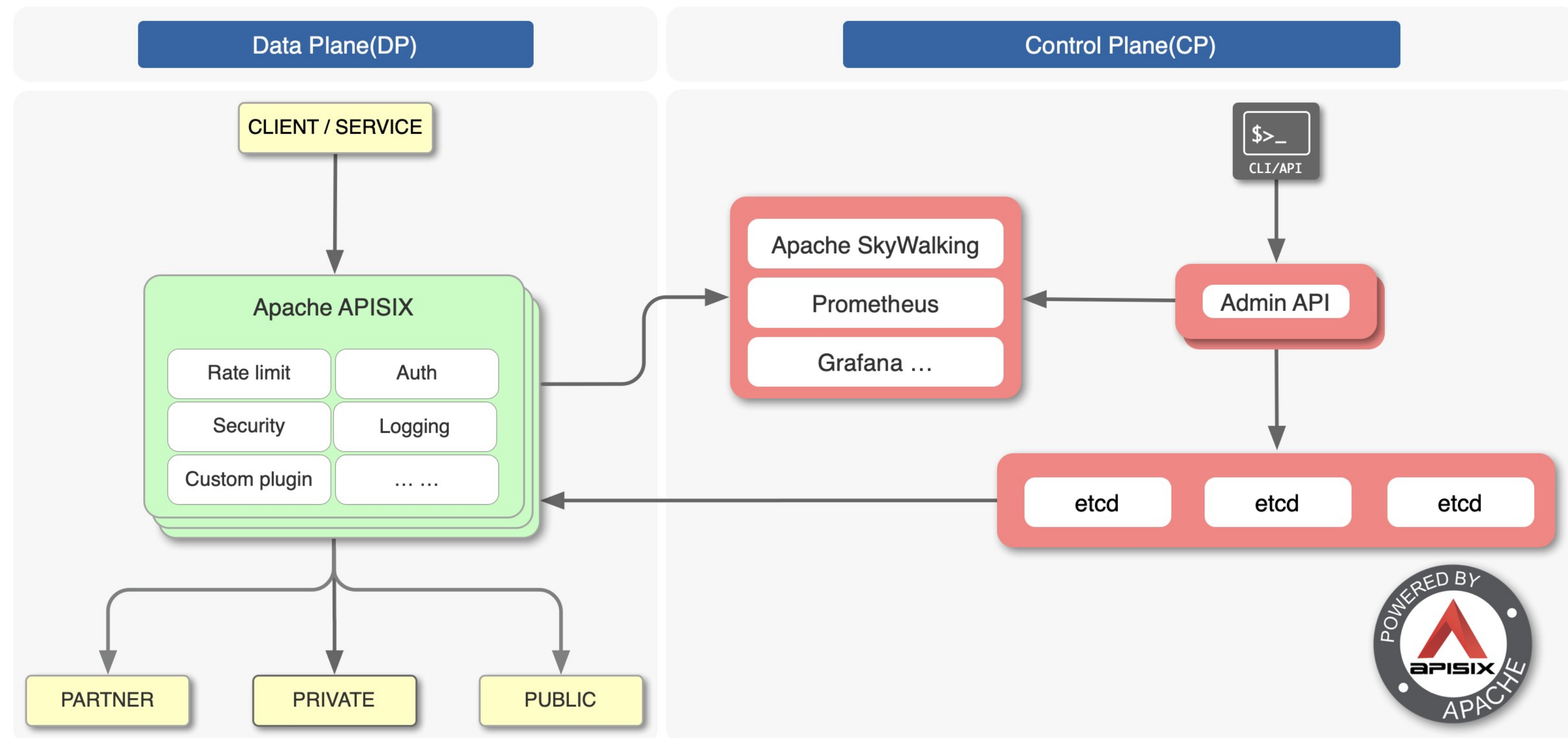
- 另一种 Ingress Controller 实现
- 使用 Apache APISIX 承载业务流量
- 通过 CRD 进行实现
- 支持原生 Ingress 资源



Apache APISIX 是什么



- Apache 基金会顶级项目
- 最活跃的开源网关项目
- 全动态高性能网关
- 云原生网关



Apache APISIX Ingress 特性

- 全动态：路由、SSL 证书、上游、插件...
- 支持 Custom Resource Definitions ，更容易理解的声明式配置
- 支持 K8S 原生 Ingress 配置（v1 / v1beta1）
- 通过 annotation 的方式对 Ingress 能力进行扩展
- 服务自动注册发现，无惧扩缩容
- 更灵活的负载均衡策略
- 健康检查开箱即用

Apache APISIX Ingress 特性



- 支持高级路由匹配规则
- 支持流量切分
- 支持 Apache APISIX 官方 50 多个插件 & 客户自定义插件
- gRPC plaintext 支持
- TCP 4 层代理
- 状态检查：快速掌握声明配置的同步状态
- ...

APISIX Ingress vs K8s ingress-nginx

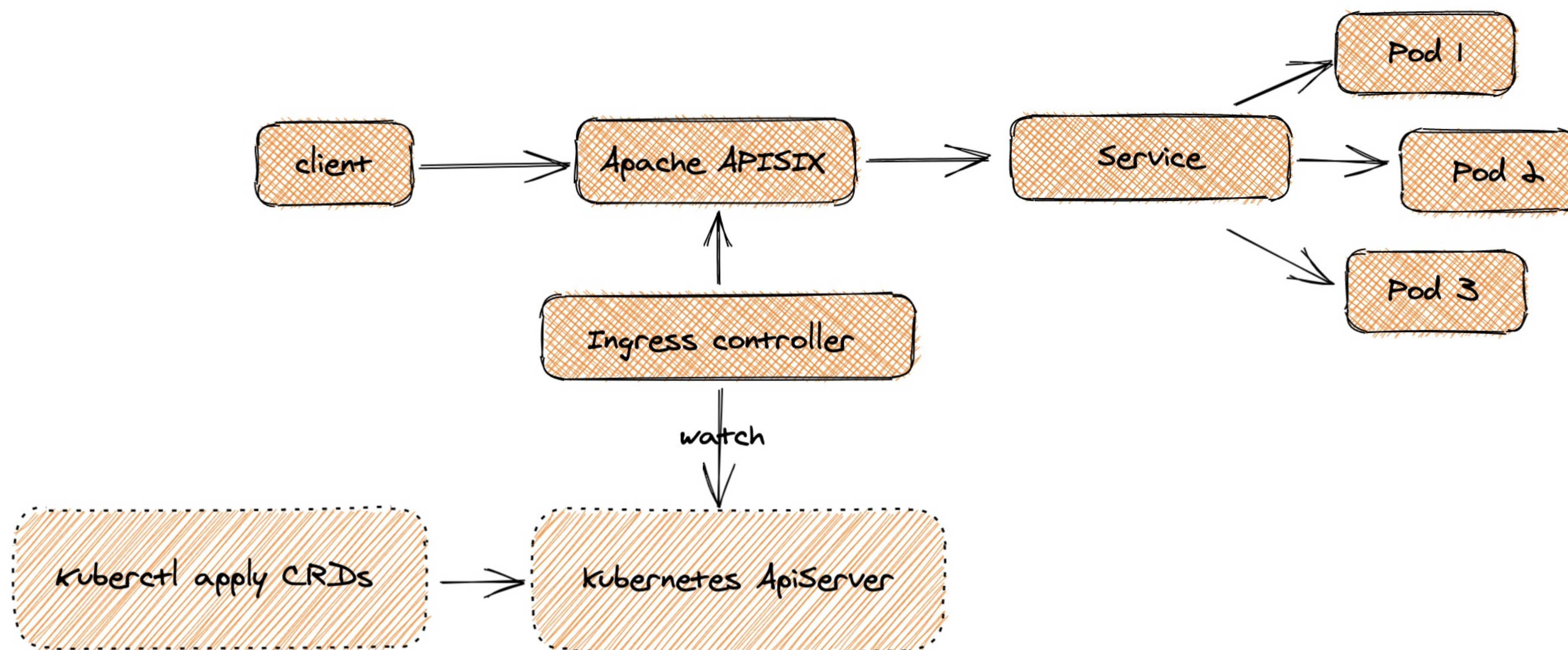


- 单个配置实现灰度部署
- 丰富的插件和功能
- 数据面和控制面架构分离

概览



通过 CRD 进行扩展



自定义资源

- ApisixRoute: 路由
- ApisixUpstream: 上游
- ApisixConsumer: 消费者
- ApisixTls: 证书相关
- ApisixClusterConfig: 集群公共配置
- ApisixPluginConfig: 插件配置 (PR 中)

ApisixRoute

- Spec.http → 7 层代理
- Spec.stream → 4 层代理

```
1 apiVersion: apisix.apache.org/v2beta1
2 kind: ApisixRoute
3 metadata:
4   name: httpbin-route
5 spec:
6   http:
7     - name: rule1
8       match:
9         hosts:
10          - httpbin.org
11         paths:
12          - /ip
13         exprs:
14          - subject:
15              scope: Header
16              name: X-Foo
17              op: Equal
18              value: bar
19       backends:
20         - serviceName: httpbin
21           servicePort: 80
22           weight: 50
23         - serviceName: httpbin
24           servicePort: 80
25           weight: 50
```



ApisixUpstream

- name 与 backend (svc) 一致
- Spec.loadbalancer : 负载均衡策略
- Spec.scheme : 协议, 支持 http 和 grpc
- Spec.healthCheck : 自定义健康检查

```
1 apiVersion: apisix.apache.org/v1
2 kind: ApisixUpstream
3 metadata:
4   name: httpbin
5 spec:
6   loadbalancer:
7     type: ewma
8   scheme: http
9   retries: 3
10  healthCheck:
11    active:
12      type: http
13      httpPath: /healthz
14      healthy:
15        interval: 3s
16        httpCodes: 200
17        successes: 2
```

ApisixConsumer



- Spec.authParameter : 支持 basicAuth 和 keyAuth
- 支持直接配置或使用 secret

```
1 apiVersion: apisix.apache.org/v2alpha1
2 kind: ApisixConsumer
3 metadata:
4   name: baz-consumer
5 spec:
6   authParameter:
7     basicAuth:
8       value:
9         username: foo
10        password: bar
```


ApisixTls



- Spec.hosts : 域名
- Spec.secret : 证书
- Spec.client : 用于 mTLS

```
1 apiVersion: apisix.apache.org/v1
2 kind: ApisixTls
3 metadata:
4   name: sample-tls
5 spec:
6   hosts:
7     - tls.httpbin.local
8   secret:
9     name: server-secret
10  namespace: default
```

ApisixClusterConfig



➤ 用于通用公共配置

```
1 apiVersion: apisix.apache.org/v2alpha1
2 kind: ApisixClusterConfig
3 metadata:
4   name: default
5 spec:
6   monitoring:
7     prometheus:
8       enable: true
```


ApisixPluginConfig



- 作为通用 Plugin 配置的集合
- 正在设计中
- <https://github.com/apache/apisix-ingress-controller/pull/689>

三、Apache APISIX Ingress 的上手实践

安装



```
1 → ~ helm install apisix apisix/apisix --set gateway.type=NodePort --set ingress-  
  controller.enabled=true --namespace apisix  
2 ...  
3 NAME: apisix  
4 LAST DEPLOYED: Fri Sep 23 16:48:38 2021  
5 NAMESPACE: apisix  
6 STATUS: deployed  
7 REVISION: 1  
8 TEST SUITE: None  
9 NOTES:  
10 1. Get the application URL by running these commands:  
11   export NODE_PORT=$(kubectl get --namespace apisix -o jsonpath="  
    {.spec.ports[0].nodePort}" services apisix-gateway)  
12   export NODE_IP=$(kubectl get nodes --namespace apisix -o jsonpath="  
    {.items[0].status.addresses[0].address}")  
13   echo http://$NODE_IP:$NODE_PORT
```


流量切分



➤ 按比例进行流量切割

```
1 apiVersion: apisix.apache.org/v1
2 kind: ApisixUpstream
3 metadata:
4   name: httpbin
5 spec:
6   subsets:
7     - name: v1
8       labels:
9         version: v1
10    - name: v2
11      labels:
12        version: v2
```

```
1 apiVersion: apisix.apache.org/v2alpha1
2 kind: ApisixRoute
3 metadata:
4   name: httpbin-route
5 spec:
6   http:
7     - name: rule1
8       match:
9         hosts:
10          - local.httpbin.com
11         paths:
12          - /ip
13       backends:
14         - serviceName: httpbin
15           servicePort: 80
16           weight: 90
17           subset: v1
18         - serviceName: httpbin
19           servicePort: 80
20           weight: 10
21           subset: v2
22
```


Basic Auth

➤对携带指定 header 访问路由的请求增加 basic auth

```
1 apiVersion: apisix.apache.org/v2alpha1
2 kind: ApisixConsumer
3 metadata:
4   name: basicvalue
5 spec:
6   authParameter:
7     basicAuth:
8       value:
9         username: foo
10        password: bar
```

```
1 apiVersion: apisix.apache.org/v2alpha1
2 kind: ApisixRoute
3 metadata:
4   name: httpbin-route
5 spec:
6   http:
7     - name: rule1
8       match:
9         hosts:
10          - httpbin.org
11         paths:
12          - /ip
13         exprs:
14          - subject:
15              scope: Header
16              name: X-Foo
17              op: Equal
18              value: bar
19         backend:
20           serviceName: httpbin
21           servicePort: 80
22         authentication:
23           enable: true
24           type: basicAuth
```


对 Ingress 的扩展



```
1 apiVersion: networking.k8s.io/v1
2 kind: Ingress
3 metadata:
4   annotations:
5     nginx.ingress.kubernetes.io/rewrite-target: /ip
6   name: ingress-rewrite
7 spec:
8   rules:
9   - host: httpbin.org
10     http:
11       paths:
12       - backend:
13           serviceName: httpbin
14           servicePort: 80
15         path: /sample
```

```
1 apiVersion: networking.k8s.io/v1
2 kind: Ingress
3 metadata:
4   annotations:
5     kubernetes.io/ingress.class: apisix
6     k8s.apisix.apache.org/rewrite-target: "/ip"
7   name: ingress-v1
8 spec:
9   rules:
10   - host: httpbin.org
11     http:
12       paths:
13       - path: /sample
14         pathType: Exact
15         backend:
16           service:
17             name: httpbin
18             port:
19               number: 80
```

四、Apache APISIX Ingress 后续规划

后续规划

- Kubernetes v1.22+ CRD v1 支持（PR 中）
- cert-manager 集成（已合并）
- Gateway API 支持（Q4）
- 扩展新架构
- 丰富生态，与 ArgoCD 等工具结合
- 扩展社区



- 谢谢！
 - zhangjintao@apache.org
- <https://apisix.apache.org>