



#APISIXNorthAmericaTour

Accessing Kubernetes pods

Back to basics and beyond

 @nicolas_frankel

Me, myself and I

#APISIXNorthAmericaTour



- Developer
- Developer advocate



 @nicolas_frankel

The pod object

#APISIXNorthAmericaTour



“Group of one or more containers, with shared storage and network resources, and a specification for how to run the containers”

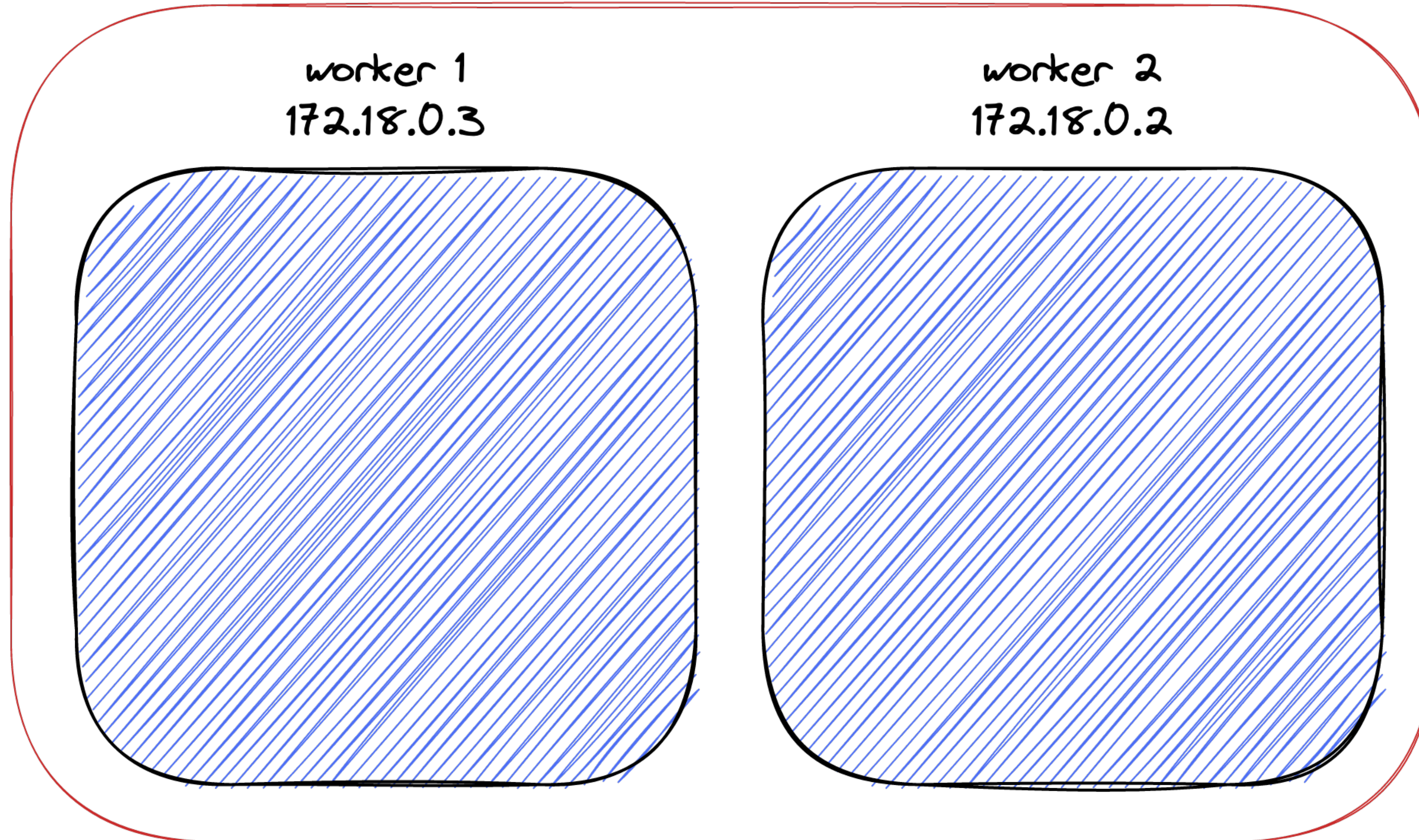
-- <https://kubernetes.io/docs/concepts/workloads/pods/>

 @nicolas_frankel



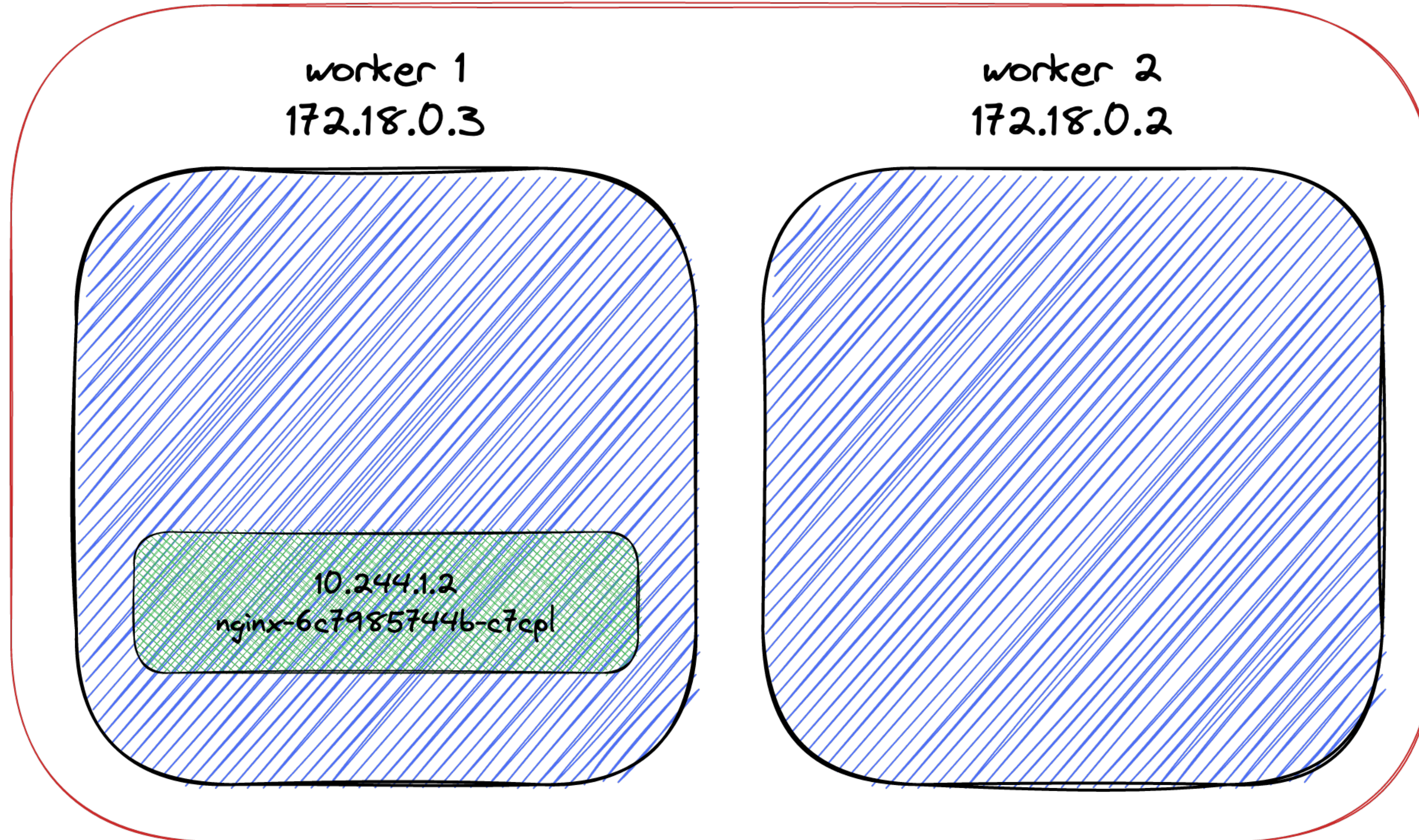
Our sample Kubernetes cluster

#APISIXNorthAmericaTour



Let's deploy our first pod

#APISIXNorthAmericaTour



Internal IPs are not stable

- Pods come and go
- Using a pod's IP is a risk



The Service object

#APISIXNorthAmericaTour



“Abstract way to expose an application running on a set of Pods as a network service. [...] Kubernetes gives Pods their own IP addresses and a single DNS name for a set of Pods, and can load-balance across them.”

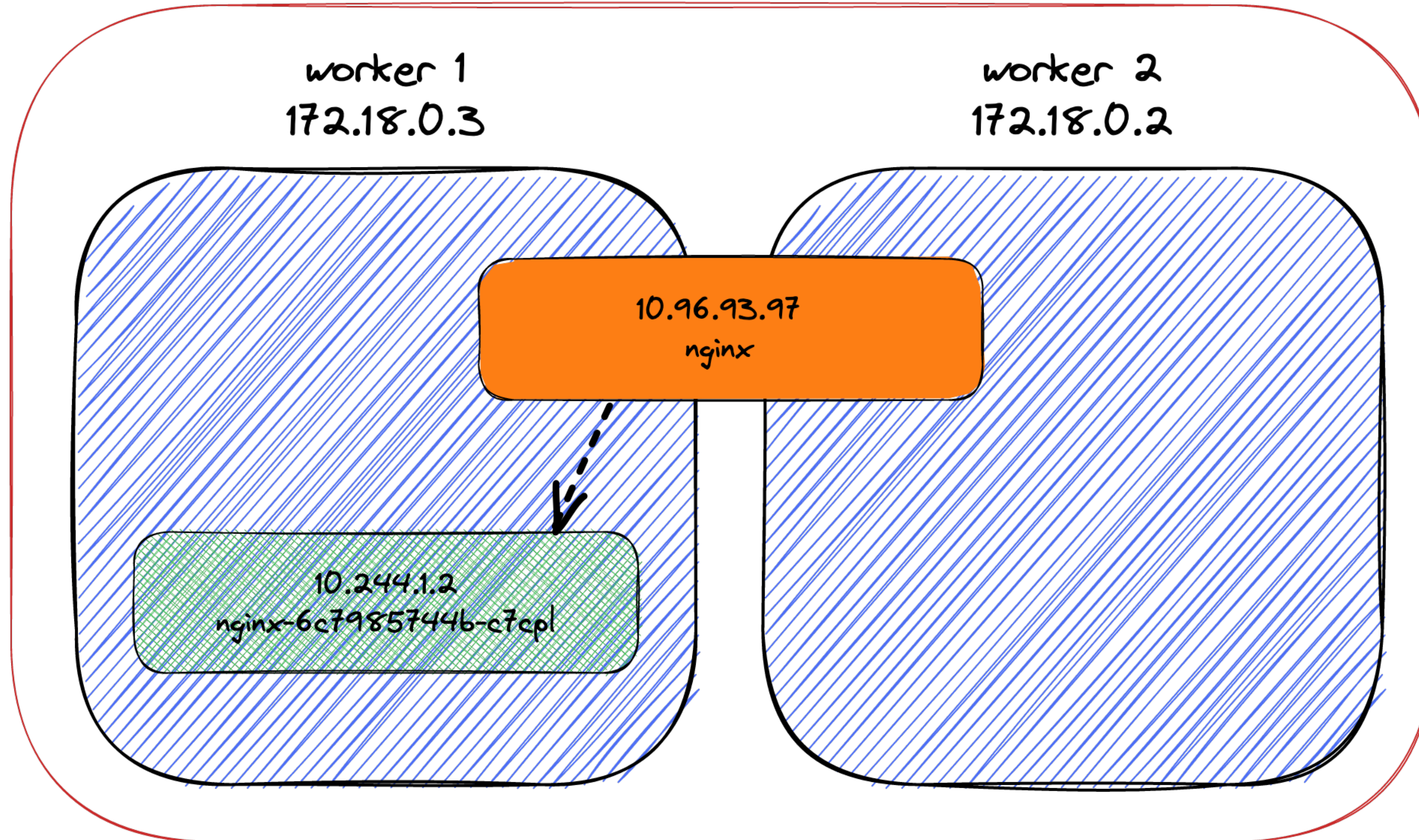
-- <https://kubernetes.io/docs/concepts/services-networking/service/>

 @nicolas_frankel



Service of type ClusterIP

#APISIXNorthAmericaTour



How to get external access?

#APISIXNorthAmericaTour

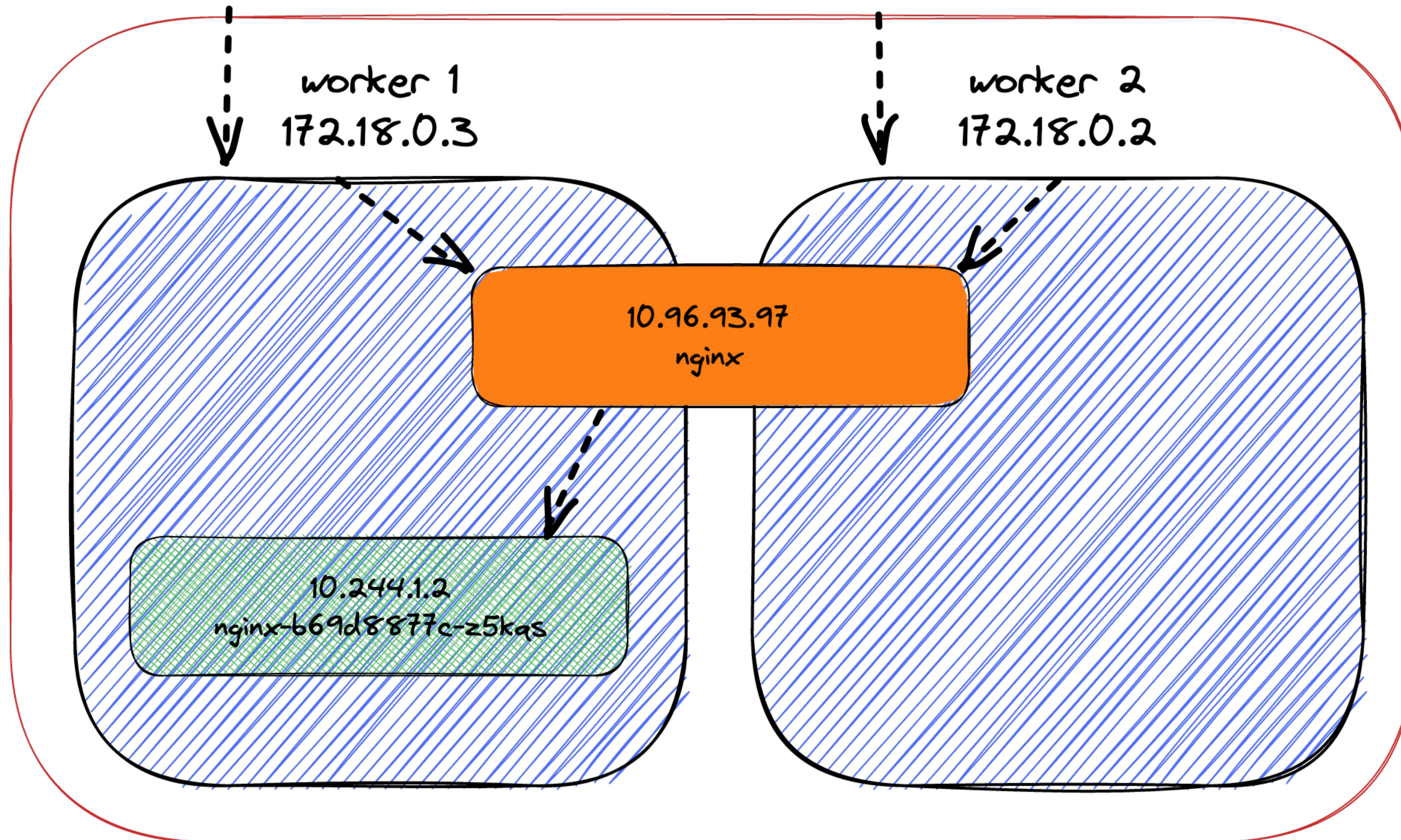


- Cluster IP is great but doesn't allow external access
- NodePort is the answer



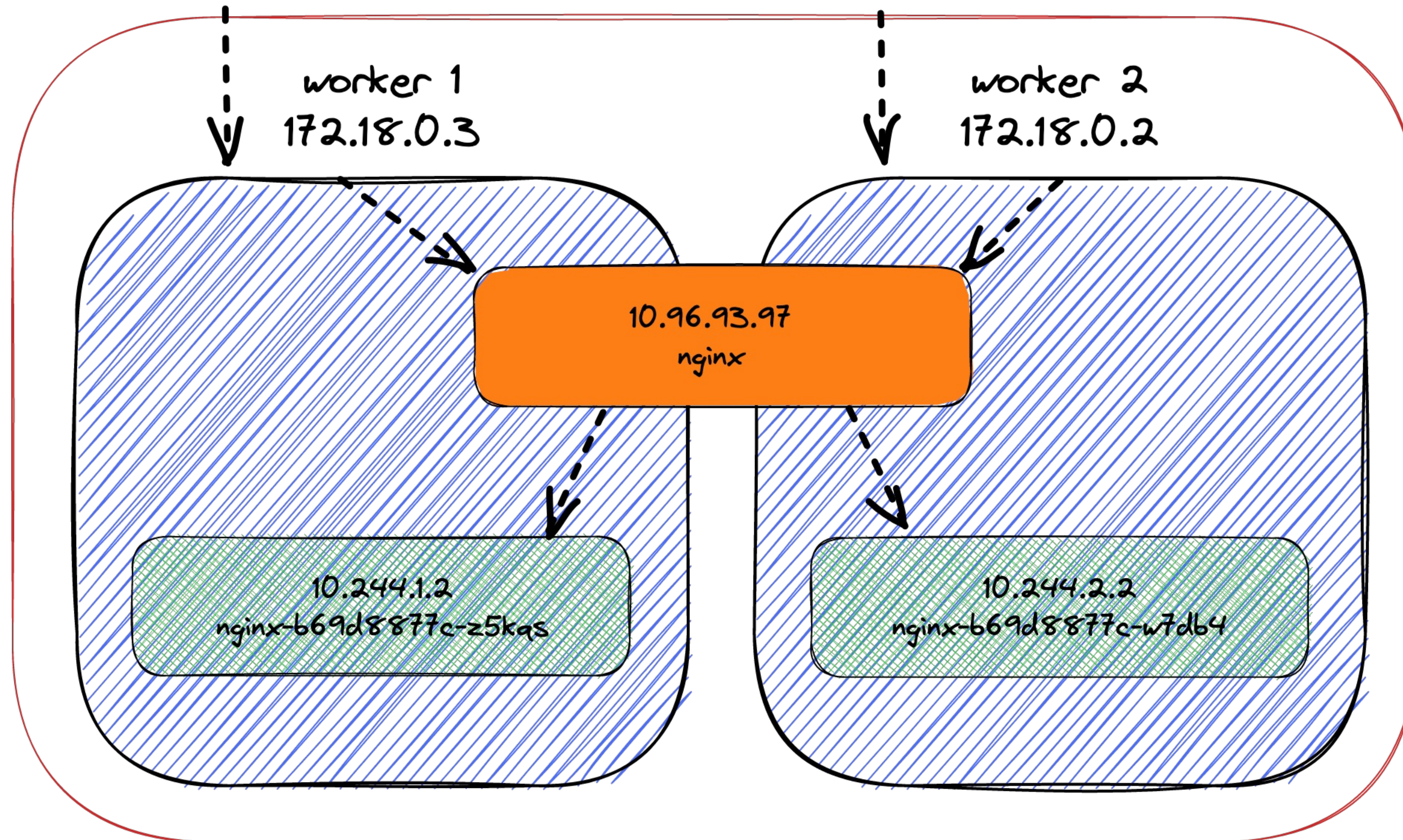
Service of type NodePort

#APISIXNorthAmericaTour



NodePort does LoadBalancing!

#APISIXNorthAmericaTour



The curious case of the Service of type LoadBalancer

#APISIXNorthAmericaTour



- When Cloud Providers want to provide additional (and proprietary) service
- It works on my machine
 - MetalLB



What about routing?

#APISIXNorthAmericaTour



- So far, pods are all the same
- What if you want to route some request to some pods?



Meet the Ingress object

#APISIXNorthAmericaTour



“Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.”

-- <https://kubernetes.io/docs/concepts/services-networking/ingress/>

 @nicolas_frankel



One abstraction to rule them all

- Apache APISIX
- Kong Gateway
- Tyk
- Gloo
- Ambassador
- HAProxy
- Nginx
- etc.

 @nicolas_frankel

#APISIXNorthAmericaTour



Apache APISIX

#APISIXNorthAmericaTour

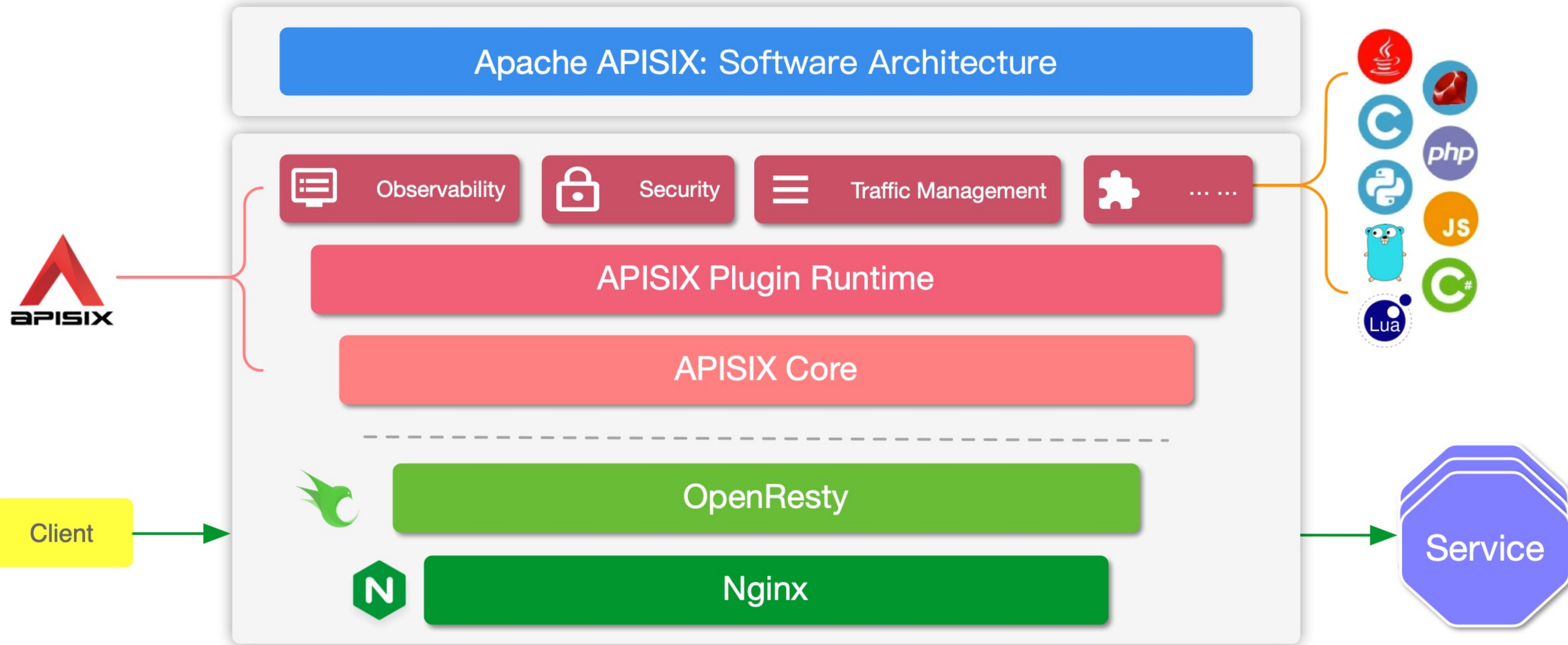


- Donated to the Apache Foundation in 2019
- Became a Top-Level Project in 2020



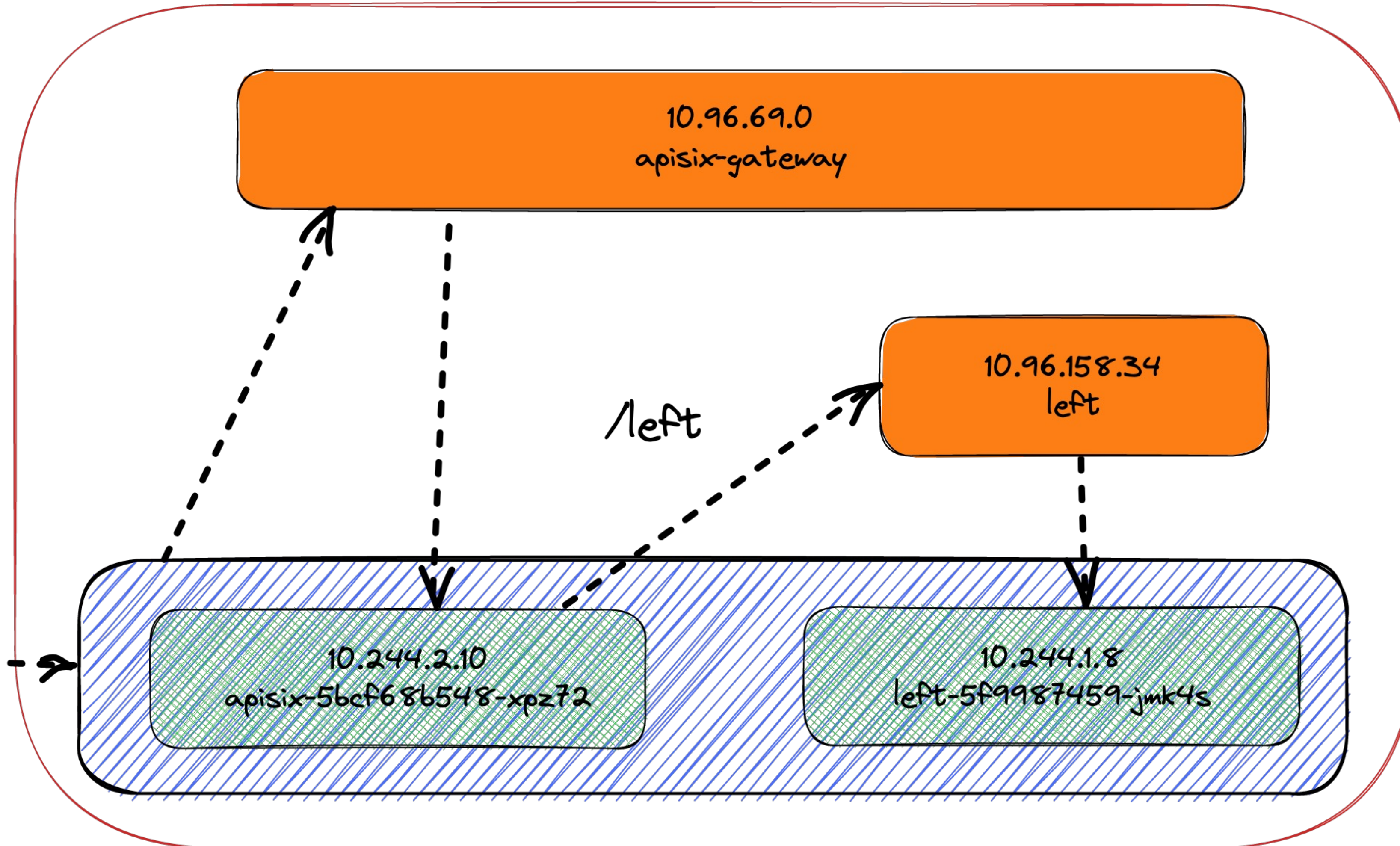
Apache APISIX, an API Gateway the Apache way

#APISIXNorthAmericaTour



Ingress

#APISIXNorthAmericaTour



Ingress

#APISIXNorthAmericaTour



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: apisix-route
spec:
  ingressClassName: apisix
  rules:
  - http:
    paths:
    - path: "/left"
      pathType: Prefix
      backend:
        service:
          name: left
          port:
            number: 80
    - path: "/right"
      pathType: Prefix
      backend:
        service:
          name: right
          port:
            number: 80
```

We rely on an existing IngressController named apisix



A simple versioning use-case

#APISIXNorthAmericaTour



■ Path based

- /v1/myapp

■ Query based

- /myapp?v=1

■ Header based

- Content-Type: application/vnd.pub.v1+json

Standard Ingress limitation

- No URL rewrites



Ingress

#APISIXNorthAmericaTour



Rewrites are not portable!

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: apisix-route
  annotations:
    k8s.apisix.apache.org/rewrite-target-regex: "[v1|v2]/(.*)"
    k8s.apisix.apache.org/rewrite-target-regex-template: "/$1"
spec:
  ingressClassName: apisix
  rules:
    - http:
      paths:
        - path: "/v1"
          pathType: Prefix
          backend:
            name: v1
            port:
              number: 80
        - path: "/v2"
          pathType: Prefix
          backend:
            name: v2
            port:
              number: 80
```



Another Ingress limitation

#APISIXNorthAmericaTour



- Path-based routing only



Custom CRD: ApisixRoute

#APISIXNorthAmericaTour



```
apiVersion: apisix.apache.org/v2
kind: ApisixRoute
metadata:
  name: apisix-route
spec:
  http:
    - name: v1
      match:
        path: ["/*"]
        exprs:
          - subject:
              scope: Query
              name: v
              op: Equal
              value: "1"
      backends:
        - serviceName: v1
          servicePort: 80
```

To bypass limitations, we use custom CRDs

Unlock the power of query-based routing



Yet another Ingress limitation

#APISIXNorthAmericaTour



- Ingress granularity
 - What if I want different teams to manage different routes?
 - They all need to work on the same Ingress object



The new kid on the block: the Gateway API

#APISIXNorthAmericaTour



- Splits the Gateway and the Routes
- Widens the scope of standard features
 - e.g. HTTP Header manipulation
 - And much much more

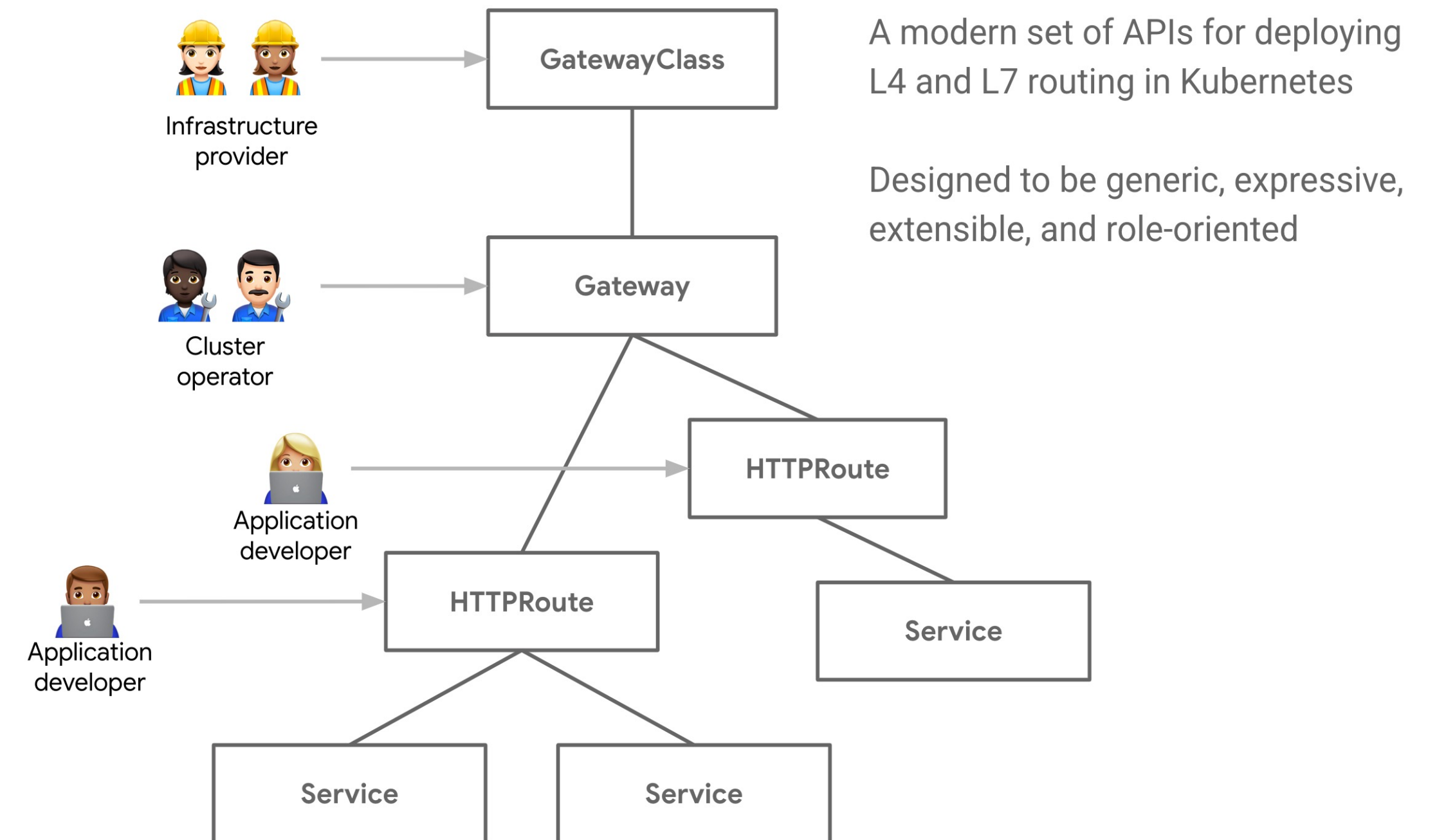


Gateway API

#APISIXNorthAmericaTour



Solve most of Ingress' issues



API versioning with Gateway API

#APISIXNorthAmericaTour



- HTTPRouteMatch
 - HTTPPathMatch
 - HTTPHeaderMatch
 - HTTPQueryParamMatch



Query-based routing

#APISIXNorthAmericaTour



```
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: dispatch
spec:
  rules:
    - matches:
      - queryParams:
          - type: Exact
            name: v
            value: 1
        backendRefs:
          - name: v1
            port: 80
```

Look, ma, no custom CRD



Routing with rewrite

#APISIXNorthAmericaTour



```
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: left
spec:
  parentRefs:
    - name: apisix-gateway
  rules:
    - filters:
      - type: URLRewrite
        urlRewrite:
          path:
            type: ReplaceFullPath
            replaceFullPath: "/rewritten"
    backendRefs:
      - name: left
        port: 80
```

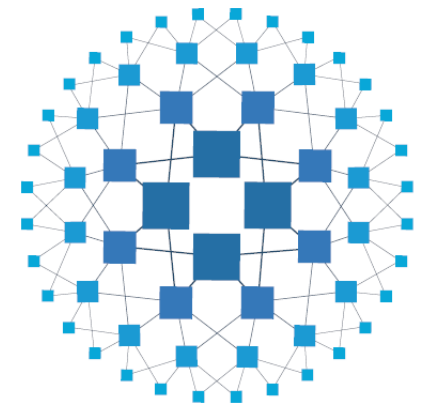




GlooEdge



#APISIXNorthAmericaTour



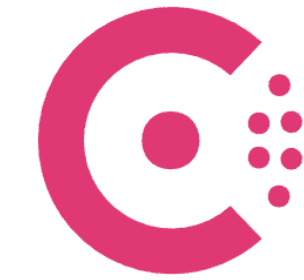
HAPROXY



APISIX



CONTOUR



HashiCorp

Consul



Kuma



LITESPEED



cilium

traefik



Kong



EMISSARY INGRESS

@nicolas_frankel

Support is implementation dependent

#APISIXNorthAmericaTour



■ For Apache APISIX

- RewriteRule isn't part of Gateway CRDs v0.5
 - Not supported yet*
- No GatewayClass, no Gateway

*<https://github.com/apache/apisix-ingress-controller/issues/1418#issuecomment-1314970927>

 @nicolas_frankel



Be careful of the Gateway API (as of now)

#APISIXNorthAmericaTour





- ~~Evolving specification~~
 - Released as of Nov. 2023
- Evolving implementations
- Gateway limitations still being discovered
- One spec to rule all implementations is simplistic



Thanks for your attention!

#APISIXNorthAmericaTour



- @nicolas_frankel 
- @nico@frankel.ch 
- nfrankel.bsky.social 
- <https://apisix.apache.org/>



 @nicolas_frankel