Calico Gateway API

Month Year

Presented by *Solutions Architect*

TIGERA

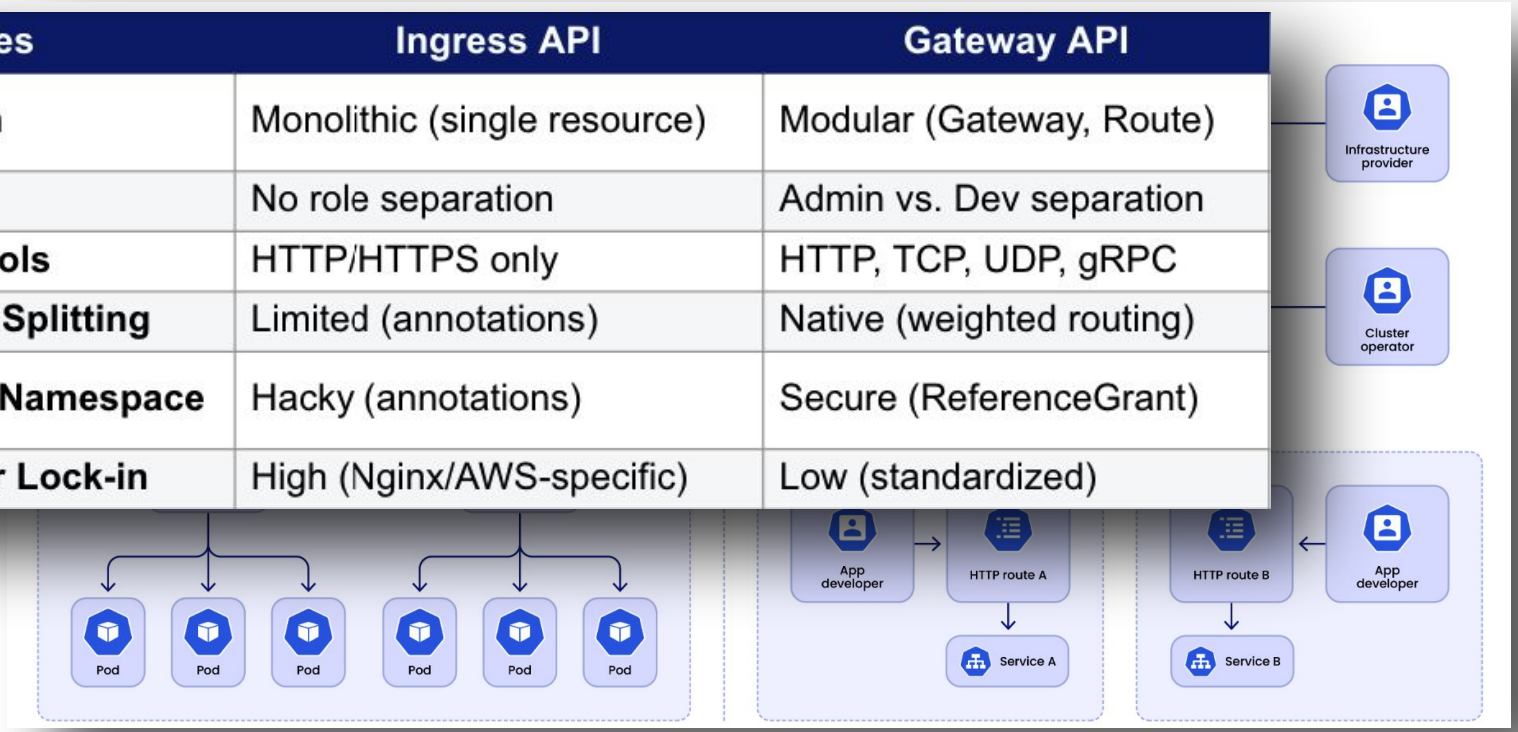
Agenda

- Legacy Ingress API vs Gateway API
- Ingress2gateway & Challenges
- Example of migration

TIGERA

Legacy Ingress API vs Gateway API

Features	Ingress API	Gateway API
Design	Monolithic (single resource)	Modular (Gateway, Route)
Roles	No role separation	Admin vs. Dev separation
Protocols	HTTP/HTTPS only	HTTP, TCP, UDP, gRPC
Traffic Splitting	Limited (annotations)	Native (weighted routing)
Cross-Namespace	Hacky (annotations)	Secure (ReferenceGrant)
Vendor Lock-in	High (Nginx/AWS-specific)	Low (standardized)



TIGERA

Ingress2gateway

- Helps translate Ingress and provider-specific resources to Gateway API resources
  - Reads resources from a Kubernetes cluster or a file
  - Outputs the equivalent Gateway API resources in a YAML/JSON format to stdout.
- Is managed by the Gateway API SIG-Network subproject

Ingress2gateway

- The simplest case is to convert all ingresses from one provider (i.e: **ingress-nginx**)

```
./ingress2gateway print --providers=ingress-nginx
```

- The above command will:
  a. Read your Kube config file to extract the cluster credentials and the current active namespace
  b. Search for ingress-nginx resources in that namespace.
  c. Convert them to Gateway-API resources (Currently only Gateways and HTTPRoutes).

https://github.com/kubernetes-sigs/ingress2gateway/blob/main/README.md

Ingress2gateway & Challenges

ingress2gateway tool

It simplifies the migration by transforming Ingress resources into Gateway API resources that Envoy Gateway can use

Ingress VS GatewayAPI

Ingress features and options are usually configured via annotations (vendor lock-in)

GatewayAPI features and options are managed by CRDs

Challenges

Not all annotations can be translated by the tool

Some options, for example OIDC integration, require a manual translation

Ingress2gateway & Challenges

Ingress2gateway supports translating these ingress-nginx specific annotations

- `nginx.ingress.kubernetes.io/canary`: If set to true will enable weighting backends.
- `nginx.ingress.kubernetes.io/canary-by-header`: If specified, the value of this annotation is the header name that will be added as a HTTPHeaderMatch for the routes generated from this Ingress. If not specified, no HTTPHeaderMatch will be generated.
- `nginx.ingress.kubernetes.io/canary-by-header-value`: If specified, the value of this annotation is the header value to perform an HeaderMatchExact match on in the generated HTTPHeaderMatch.
- `nginx.ingress.kubernetes.io/canary-by-header-pattern`: If specified, this is the pattern to match against for the HTTPHeaderMatch, which will be of type HeaderMatchRegularExpression.
- `nginx.ingress.kubernetes.io/canary-weight`: If specified and non-zero, this value will be applied as the weight of the backends for the routes generated from this Ingress resource.
- `nginx.ingress.kubernetes.io/canary-weight-total`

https://github.com/kubernetes-sigs/ingress2gateway/blob/main/pkg/i2gw/providers/ingressnginx/README.md

Sample of NGINX Ingress

- Canary deployment (will be translated in GatewayAPI **HTTPRoute**)
- OIDC Authentication (will be manually converted in GatewayAPI **SecurityPolicy**)

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-app
  namespace: default
  annotations:
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/auth-url: "http://oauth2-proxy.default.svc.cluster.local:4180/oauth2/auth"
    nginx.ingress.kubernetes.io/auth-signin: "https://$host/oauth2/start?rd=$request_uri"
    nginx.ingress.kubernetes.io/canary: "true"
    nginx.ingress.kubernetes.io/canary-weight: "20"
```

Sample of GatewayAPI resources

Gateway

```
apiVersion:
gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: nginx-to-cig
  namespace: default
spec:
  gatewayClassName:
tigera-gateway-class
  listeners:
  - hostname:
echoserver.myingress.com
    name:
echoserver-myingress-com-http
    port: 80
    protocol: HTTP
```

HTTPRoute

```
apiVersion:
gateway.networking.k8s.io/v1
kind: HTTPRoute
...
  rules:
  - backendRefs:
    - name: app-v1
      port: 8080
      weight: 80
    - name: app-v2
      port: 8080
      weight: 20
    matches:
    - path:
        type: PathPrefix
        value: /
```

SecurityPolicy

```
apiVersion:
gateway.envoyproxy.io/v1alpha1
kind: SecurityPolicy
...
spec:
  oidc:
    provider:
      issuer:
"https://accounts.go……"
      clientSecretRef:
        name: oidc-client-secret
        key: client-secret
    authorizationEndpoint: "https:..."
    tokenEndpoint: "https://oaut…."
    userInfoEndpoint: "https://op..."
    redirectURI: "https://myapp.e..."
```

Summary of Sample Migration

NGINX Ingress	Calico Ingress Gateway
Vendor specific	Standard k8s GatewayAPI
Number of resources: **2** ingress for app-v1 ingress for app-v2 (canary)	Number of resources: **3** Gateway HTTPRoute SecurityPolicy
Kind: Ingress	Kind: Gateway
“Canary” annotations	Kind: HTTPRoute
“OIDC” annotations	Kind: SecurityPolicy

TIGERA

Thank you

TIGERA

Follow us on: