



Unleashing the Future of Kubernetes Networking

(How Gateway API Transforms Container Exposure)

Emile Vauge

Traefik *Creator*

Traefik Labs *CTO & Founder*



Nicolas Mengin

Traefik *Maintainer*

Traefik Labs *Head of Support*

Why a new Specification?



Ingress Evolution

Ingress (2015)

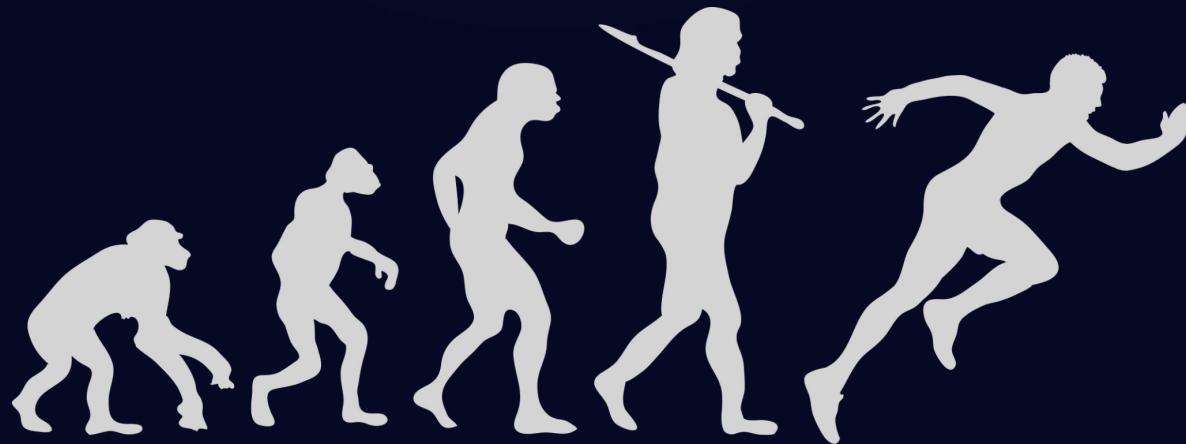
Vendor Neutral (Theoretically)
Specific Configuration using Annotations
Only HTTP
Deprecated

IngressRoute (2019)

Vendor Specific
Configuration using *spec*
HTTP/TCP/UDP
Middlewares CRD

Gateway API (2019)

Vendor Neutral (Expected)
Configuration using objects
HTTP/TCP
Next Standard
Active Roadmap
Ingress and Mesh



Ingress ❤️ Simplicity

- 😊 HTTP routes
- 😊 host, *, path & prefix rules
- 😊 default backend
- 😊 TLS



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  namespace: default
  annotations:
    kubernetes.io/ingress.class: "nginx"
spec:
  tls:
  - hosts:
    - example.com
    secretName: example-tls
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
      backend:
        service:
          name: example-service
        port:
          number: 80
```



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  namespace: default
  annotations:
    kubernetes.io/ingress.class: "traefik"
spec:
  tls:
  - hosts:
    - example.com
    secretName: example-tls
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
      backend:
        service:
          name: example-service
        port:
          number: 80
```

Ingress ❤️ Simplicity 💔 Reality

- | | |
|--|---|
| 😊 HTTP routes | 😢 Rate limiting / Retries / Canary |
| 😊 host, *, path & prefix rules | 😢 Authentication (OAuth, JWT, etc.) |
| 😊 default backend | 😢 CORS policies |
| 😊 TLS | 😢 Headers/Request/Response modification |
| 😢 TCP, UDP | 😢 Multi-tenancy / Namespace Isolation |
| 😢 mTLS | 😢 Separation of concerns |
| 😢 Adv Matching Rules (Method, OR/AND, Regex) | 😢 Extensibility |

WORKAROUND?



```
nginx.ingress.kubernetes.io/app-root
nginx.ingress.kubernetes.io/affinity
nginx.ingress.kubernetes.io/affinity-mode
nginx.ingress.kubernetes.io/affinity-canary-behavior
nginx.ingress.kubernetes.io/auth-realm
nginx.ingress.kubernetes.io/auth-secret
nginx.ingress.kubernetes.io/auth-secret-type
nginx.ingress.kubernetes.io/auth-type
nginx.ingress.kubernetes.io/auth-tls-secret
nginx.ingress.kubernetes.io/auth-tls-secret-depth
nginx.ingress.kubernetes.io/auth-tls-verify-client
nginx.ingress.kubernetes.io/auth-tls-error-page
nginx.ingress.kubernetes.io/auth-tls-pass-certificate-to-upstream
nginx.ingress.kubernetes.io/auth-tls-match-cn
nginx.ingress.kubernetes.io/auth-url
nginx.ingress.kubernetes.io/auth-cache-key
nginx.ingress.kubernetes.io/auth-cache-duration
nginx.ingress.kubernetes.io/auth-keepalive
nginx.ingress.kubernetes.io/auth-keepalive-share-vars
nginx.ingress.kubernetes.io/auth-keepalive-requests
nginx.ingress.kubernetes.io/auth-keepalive-timeout
nginx.ingress.kubernetes.io/auth-proxy-set-headers
nginx.ingress.kubernetes.io/auth-snippet
nginx.ingress.kubernetes.io/enable-global-auth
nginx.ingress.kubernetes.io/backend-protocol
nginx.ingress.kubernetes.io/canary
nginx.ingress.kubernetes.io/canary-by-header
nginx.ingress.kubernetes.io/canary-by-header-value
nginx.ingress.kubernetes.io/canary-by-header-pattern
nginx.ingress.kubernetes.io/canary-by-cookie
nginx.ingress.kubernetes.io/canary-weight
nginx.ingress.kubernetes.io/canary-weight-total
nginx.ingress.kubernetes.io/client-body-buffer-size
nginx.ingress.kubernetes.io/configuration-snippet
nginx.ingress.kubernetes.io/custom-httperrors
nginx.ingress.kubernetes.io/custom-headers
nginx.ingress.kubernetes.io/default-backend
nginx.ingress.kubernetes.io/enable-cors
nginx.ingress.kubernetes.io/cors-allow-origin
nginx.ingress.kubernetes.io/cors-allow-methods
nginx.ingress.kubernetes.io/cors-allow-headers
nginx.ingress.kubernetes.io/cors-expose-headers
nginx.ingress.kubernetes.io/cors-allow-credentials
nginx.ingress.kubernetes.io/cors-max-age
nginx.ingress.kubernetes.io/force-ssl-redirect
nginx.ingress.kubernetes.io/from-to-www-redirect
nginx.ingress.kubernetes.io/http2-push-preload
nginx.ingress.kubernetes.io/limit-connections
nginx.ingress.kubernetes.io/limit-rps
nginx.ingress.kubernetes.io/permanent-redirect
nginx.ingress.kubernetes.io/permanent-redirect-code
nginx.ingress.kubernetes.io/temporal-redirect
nginx.ingress.kubernetes.io/temporal-redirect-code
nginx.ingress.kubernetes.io/preserve-trailing-slash
nginx.ingress.kubernetes.io/proxy-body-size
nginx.ingress.kubernetes.io/proxy-cookie-domain
nginx.ingress.kubernetes.io/proxy-cookie-path
nginx.ingress.kubernetes.io/proxy-connect-timeout
```

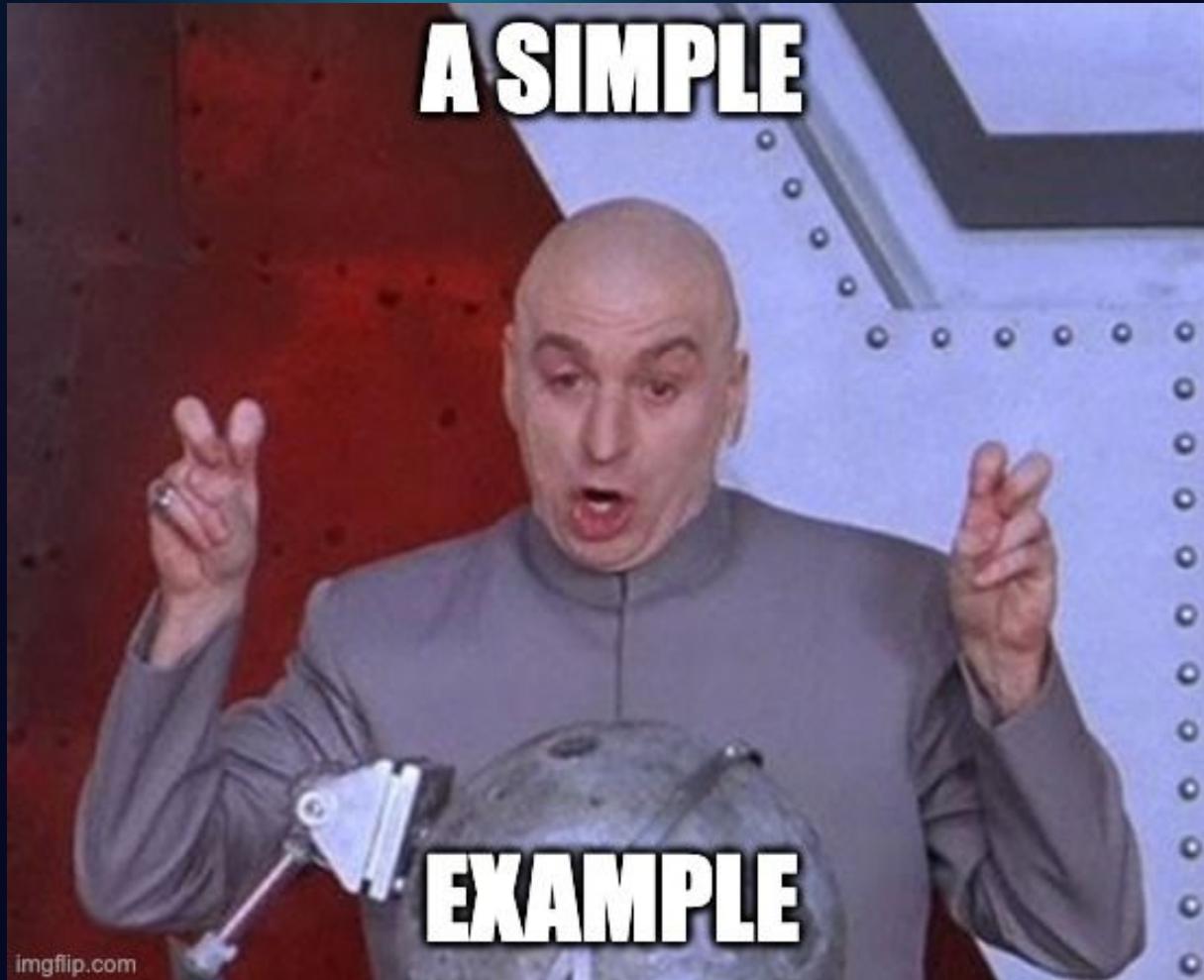


```
nginx.ingress.kubernetes.io/proxy-send-timeout
nginx.ingress.kubernetes.io/proxy-read-timeout
nginx.ingress.kubernetes.io/proxy-next-upstream
nginx.ingress.kubernetes.io/proxy-next-upstream-timeout
nginx.ingress.kubernetes.io/proxy-next-upstream-tries
nginx.ingress.kubernetes.io/proxy-request-buffering
nginx.ingress.kubernetes.io/proxy-redirect-from
nginx.ingress.kubernetes.io/proxy-redirect-to
nginx.ingress.kubernetes.io/proxy-http-version
nginx.ingress.kubernetes.io/proxy-ssl-secret
nginx.ingress.kubernetes.io/proxy-ssl-ciphers
nginx.ingress.kubernetes.io/proxy-ssl-name
nginx.ingress.kubernetes.io/proxy-ssl-protocols
nginx.ingress.kubernetes.io/proxy-ssl-verify
nginx.ingress.kubernetes.io/proxy-ssl-verify-depth
nginx.ingress.kubernetes.io/proxy-ssl-server-name
nginx.ingress.kubernetes.io/enable-rewrite-log
nginx.ingress.kubernetes.io/rewrite-target
nginx.ingress.kubernetes.io/satisfy
nginx.ingress.kubernetes.io/server-alias
nginx.ingress.kubernetes.io/server-snippet
nginx.ingress.kubernetes.io/service-upstream
nginx.ingress.kubernetes.io/session-cookie-change-on-failure
nginx.ingress.kubernetes.io/session-cookie-conditional-samesite-none
nginx.ingress.kubernetes.io/session-cookie-domain
nginx.ingress.kubernetes.io/session-cookie-expires
nginx.ingress.kubernetes.io/session-cookie-max-age
nginx.ingress.kubernetes.io/session-cookie-name
nginx.ingress.kubernetes.io/session-cookie-path
nginx.ingress.kubernetes.io/session-cookie-samesite
nginx.ingress.kubernetes.io/session-cookie-secure
nginx.ingress.kubernetes.io/ssl-redirect
nginx.ingress.kubernetes.io/ssl-passthrough
nginx.ingress.kubernetes.io/stream-snippet
nginx.ingress.kubernetes.io/upstream-hash-by
nginx.ingress.kubernetes.io/x-forwarded-prefix
nginx.ingress.kubernetes.io/load-balance
nginx.ingress.kubernetes.io/upstream-vhost
nginx.ingress.kubernetes.io/denylist-source-range
nginx.ingress.kubernetes.io/whitelist-source-range
nginx.ingress.kubernetes.io/proxy-buffering
nginx.ingress.kubernetes.io/proxy-buffers-number
nginx.ingress.kubernetes.io/proxy-buffer-size
nginx.ingress.kubernetes.io/proxy-busy-buffers-size
nginx.ingress.kubernetes.io/proxy-max-temp-file-size
nginx.ingress.kubernetes.io/ssl-ciphers
nginx.ingress.kubernetes.io/ssl-prefer-server-ciphers
nginx.ingress.kubernetes.io/connection-proxy-header
nginx.ingress.kubernetes.io/enable-access-log
nginx.ingress.kubernetes.io/enable-opentelemetry
nginx.ingress.kubernetes.io/opentelemetry-trust-incoming-span
nginx.ingress.kubernetes.io/use-regex
nginx.ingress.kubernetes.io/enable-modsecurity
nginx.ingress.kubernetes.io/enable-owasp-core-rules
nginx.ingress.kubernetes.io/modsecurity-transaction-id
nginx.ingress.kubernetes.io/modsecurity-snippet
nginx.ingress.kubernetes.io/mirror-request-body
nginx.ingress.kubernetes.io/mirror-target
nginx.ingress.kubernetes.io/mirror-host
```

1117



mTLS
sticky sessions
CORS





mTLS
sticky sessions
CORS



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: advanced-secure-ingress
annotations:
  # Configuration mTLS
  nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"
  nginx.ingress.kubernetes.io/auth-tls-secret: "default/ca-secret"
  nginx.ingress.kubernetes.io/auth-tls-verify-depth: "1"
  nginx.ingress.kubernetes.io/auth-tls-pass-certificate-to-upstream: "true"
```



mTLS
sticky sessions
CORS



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: advanced-secure-ingress
annotations:
  # Configuration mTLS
  nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"
  nginx.ingress.kubernetes.io/auth-tls-secret: "default/ca-secret"
  nginx.ingress.kubernetes.io/auth-tls-verify-depth: "1"
  nginx.ingress.kubernetes.io/auth-tls-pass-certificate-to-upstream: "true"

  # Configuration sticky sessions
  nginx.ingress.kubernetes.io/affinity: "cookie"
  nginx.ingress.kubernetes.io/session-cookie-name: "STICKYID"
  nginx.ingress.kubernetes.io/session-cookie-expire: "86400"
  nginx.ingress.kubernetes.io/session-cookie-max-age: "86400"
  nginx.ingress.kubernetes.io/session-cookie-path: "/"
```



mTLS
sticky sessions
CORS

```
...  
  
# Configuration sticky sessions  
nginx.ingress.kubernetes.io/affinity: "cookie"  
nginx.ingress.kubernetes.io/session-cookie-name: "STICKYID"  
nginx.ingress.kubernetes.io/session-cookie-expires: "86400"  
nginx.ingress.kubernetes.io/session-cookie-max-age: "86400"  
nginx.ingress.kubernetes.io/session-cookie-path: "/"  
  
# Configuration CORS  
nginx.ingress.kubernetes.io/enable-cors: "true"  
nginx.ingress.kubernetes.io/cors-allow-origin:  
"https://trusted-origin.com, https://*.example.org"  
nginx.ingress.kubernetes.io/cors-allow-methods: "GET, POST"
```



mTLS
sticky sessions
CORS

```
...  
  
# Configuration CORS  
nginx.ingress.kubernetes.io/enable-cors: "true"  
nginx.ingress.kubernetes.io/cors-allow-origin:  
"https://trusted-origin.com, https://*.example.org"  
nginx.ingress.kubernetes.io/cors-allow-methods: "GET, POST"  
  
spec:  
  ingressClassName: nginx  
  tls:  
    - hosts:  
        - api.example.com  
        secretName: tls-secret  
    rules:  
      - host: api.example.com  
        http:  
          paths:  
            - path: /  
              pathType: Prefix  
              backend:  
                service:  
                  name: api-service  
                  port:  
                    number: 8443
```



mTLS sticky sessions CORS

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: advanced-secure-ingress
annotations:
  # Specify Traefik as the ingress controller
  kubernetes.io/ingress.class: "traefik"

  # mTLS Configuration
  traefik.ingress.kubernetes.io/router.tls: "true"
  traefik.ingress.kubernetes.io/router.tls.options: "default-mtls-options@kubernetescrd"

  # CORS Configuration
  traefik.ingress.kubernetes.io/router.middlewares: "default-cors-headers@kubernetescrd"

  # Sticky Sessions
  traefik.ingress.kubernetes.io/service.sticky: "true"
  traefik.ingress.kubernetes.io/service.sticky.cookie.name: "STICKYID"
  traefik.ingress.kubernetes.io/service.sticky.cookie.secure: "true"
  traefik.ingress.kubernetes.io/service.sticky.cookie.httpOnly: "true"
  traefik.ingress.kubernetes.io/service.sticky.cookie.maxAge: "86400"
```



mTLS
sticky sessions
CORS

```
# Configuration TLS avec mTLS
apiVersion: traefik.containo.us/v1alpha1
kind: TLSOption
metadata:
  name: mtls-options
spec:
  clientAuth:
    secretNames:
      - ca-secret
  clientAuthType: RequireAndVerify
```



mTLS
sticky sessions
CORS

```
# Middleware pour les CORS
apiVersion: traefik.containo.us/v1alpha1
kind: Middleware
metadata:
  name: cors-middleware
spec:
  headers:
    accessControlAllowMethods:
      - GET
      - POST
    accessControlAllowOriginList:
      - https://trusted-origin.com
      - https://*.example.org
```

INGRESS IS

VENDOR NEUTRAL

Ingress Evolution

Ingress (2015)

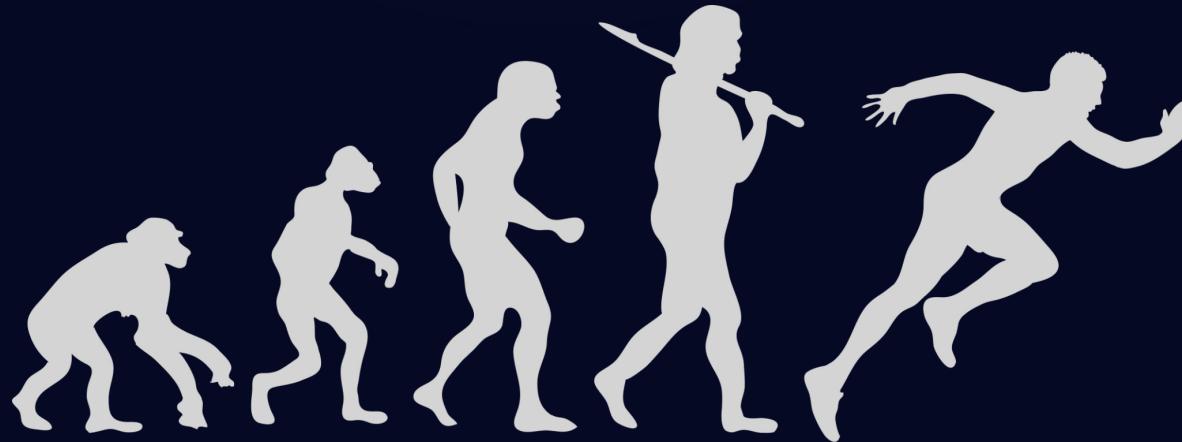
Vendor Neutral (Theoretically)
Specific Configuration using Annotations
Only HTTP
Deprecated

IngressRoute (2019)

Vendor Specific
Configuration using *spec*
HTTP/TCP/UDP
Middlewares CRD

Gateway API (2019)

Vendor Neutral (Expected)
Configuration using objects
HTTP/TCP
Next Standard
Active Roadmap
Ingress and Mesh





```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  namespace: default
  annotations:
    kubernetes.io/ingress.class: "nginx"
spec:
  tls:
  - hosts:
    - example.com
    secretName: example-tls
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: example-service
            port:
              number: 80
```



```
apiVersion: traefik.io/v1alpha1
kind: IngressRoute
metadata:
  name: example-ingressroute
  namespace: default
spec:
  entryPoints:
  - websecure
  routes:
  - match: Host(`example.com`) && PathPrefix(`/`)
    kind: Rule
    services:
    - name: example-service
      port: 80
  tls:
    secretName: example-tls
```



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  namespace: default
  annotations:
    kubernetes.io/ingress.class: "nginx"
spec:
  tls:
  - hosts:
    - example.com
    secretName: example-tls
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: example-service
            port:
              number: 80
```



```
apiVersion: projectcontour.io/v1
kind: HTTPProxy
metadata:
  name: example-httpproxy
  namespace: default
spec:
  virtualhost:
    fqdn: example.com
    tls:
      secretName: example-tls
  routes:
  - conditions:
    - prefix: /
      services:
      - name: example-service
        port: 80
```

Ingress Evolution

Ingress (2015)

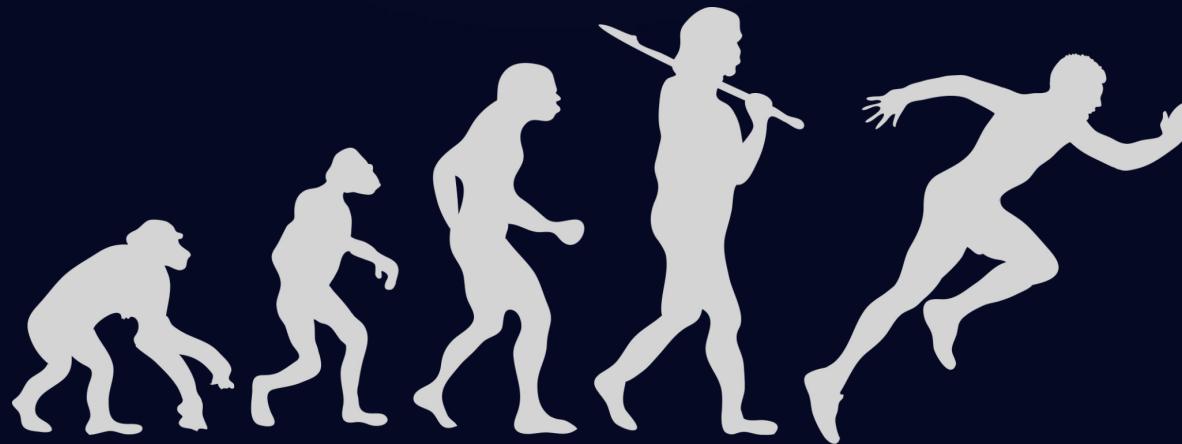
Vendor Neutral (Theoretically)
Specific Configuration using Annotations
Only HTTP
Deprecated

IngressRoute (2019)

Vendor Specific
Configuration using *spec*
HTTP/TCP/UDP
Middlewares CRD

Gateway API (2019)

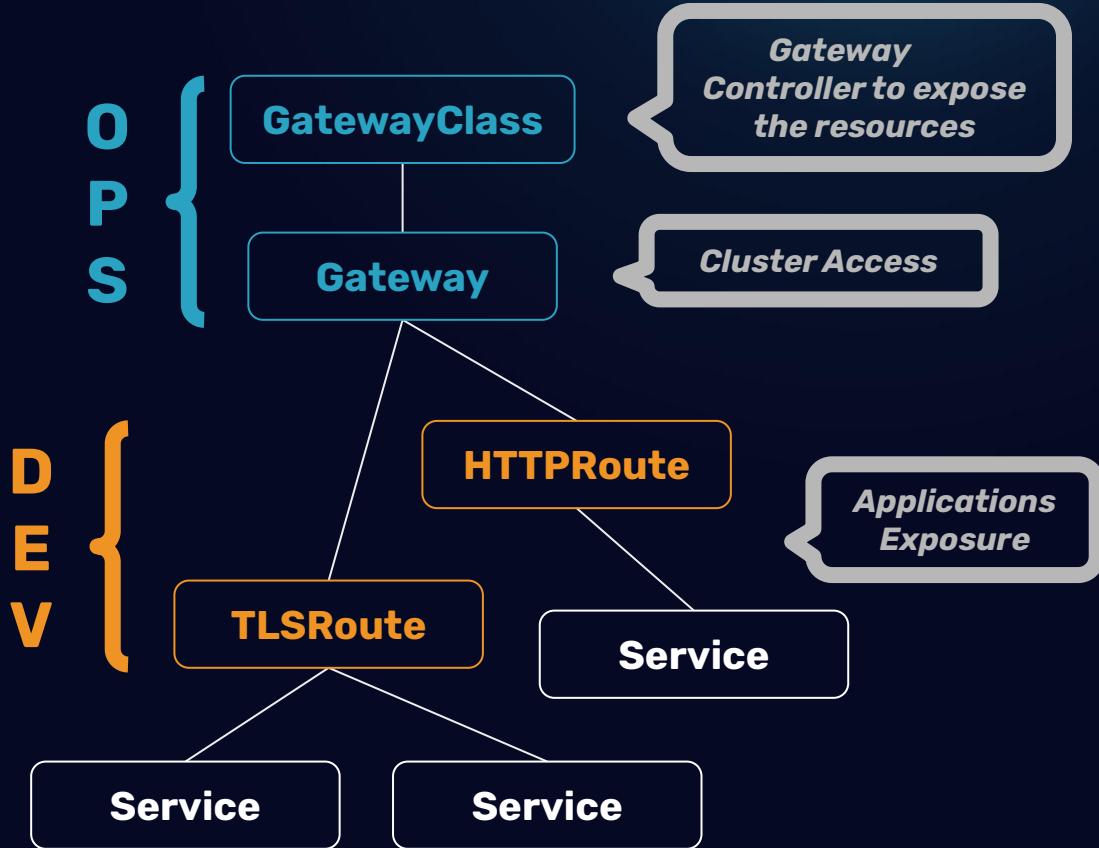
Vendor Neutral (Expected)
Configuration using objects
HTTP/TCP
Next Standard
Active Roadmap
Ingress and Mesh



Gateway API in a few words



In Short



```
# GATEWAY API #
#####
# GATEWAYCLASS
kind: GatewayClass
metadata:
  name: traefik-controller
spec:
  controllerName: "traefik.io/gateway-controller"
  ...
---

# GATEWAY
kind: Gateway
metadata:
  name: traefik-gw
spec:
  gatewayClassName: traefik-controller
  listeners:
    - name: web
      port: 443
      protocol: HTTP
      hostname: "example.com"
  ...
---

# HTTPROUTE
kind: HTTPRoute
metadata:
  name: myhttproute
spec:
  parentRefs:
    - name: traefik-gw
  hostnames:
    - example.com
  rules:
    - matches:
        ...
        filters:
          ...

```

```
● ● ●
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: myingress
  annotations:
    traefik.ingress.kubernetes.io/router.entrypoints: websecure
    traefik.ingress.kubernetes.io/router.tls: true
    traefik.ingress.kubernetes.io/router.middlewares: addHeader@kubernetescrd
spec:
  rules:
    - host: example.com
      http:
        paths:
          - path: /bar
            pathType: Exact
            backend:
              service:
                name: whoami
                port:
                  number: 80
            tls:
              - secretName: supersecret
---
apiVersion: traefik.io/v1alpha1
kind: Middleware
metadata:
  name: addHeader
spec:
  headers:
    customRequestHeaders:
      my-header-name: "my-header-value"
```

INGRESS

```
● ● ●
---
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gw
spec:
  gatewayClassName: traefik-controller
  listeners:
    - name: websecure
      port: 443
      protocol: HTTPS
      hostname: "example.com"
    tls:
      mode: Terminate
      certificateRefs:
        - kind: Secret
          name: supersecret
```

GATEWAY API

```
---  
apiVersion: gateway.networking.k8s.io/v1  
kind: HTTPRoute  
metadata:  
  name: myhttproute  
spec:  
  parentRefs:  
    - name: traefik-gw  
  hostnames:  
    - example.com  
  rules:  
    - matches:  
      - path:  
          type: Path  
          value: /bar  
    backendRefs:  
      - name: whoami  
        port: 80  
  filters:  
    - type: RequestHeaderModifier  
      requestHeaderModifier:  
        add:  
          - name: my-header-name  
            value: my-header-value
```

Let's dive in the concepts!



Ops Resources

Gateway Controller, GatewayClass

Gateway Controller

- Many Implementations



Gateway Controller

- Many Implementations
- Support Levels
 - Core
 - Extended
 - Implementation-specific

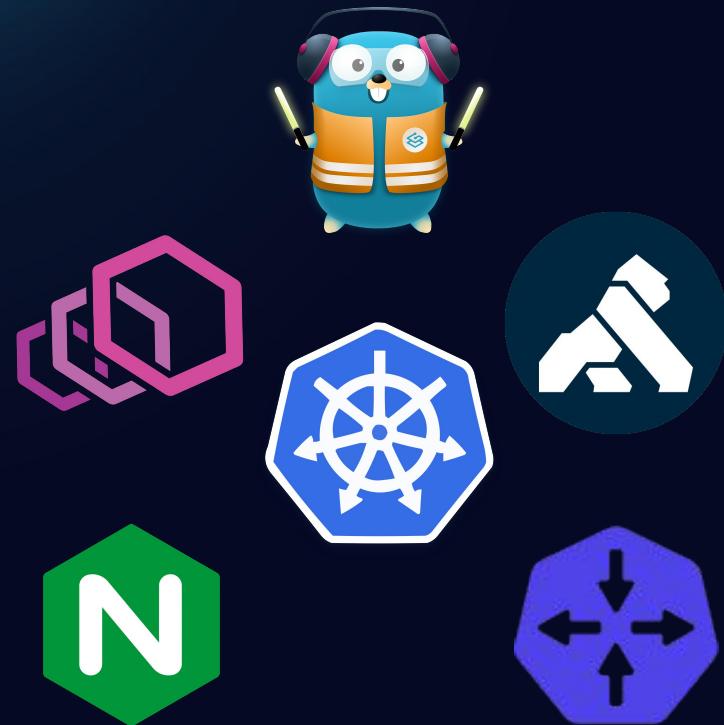


Gateway Controller

- Many Implementations
- Support Levels
- Conformance Tests

Gateway API Conformance v1.0.0 EnvoyGateway

Gateway API Conformance v1.2.1 Traefik Proxy



Gateway Controller

- Many Implementations
- Support Levels
- Conformance Tests
- Choose the best for you!





A lightweight, Gateway Controller.

Open Source • Traffic Management • Security • Observability

A screenshot of the traefik 3.3.3 dashboard. The top navigation bar includes links for Dashboard, HTTP, TCP, UDP, Plugins, an Upgrade button, Light theme, and Help. The main content area starts with a section for "Entrypoints" showing three boxes: "TRAEFIK :8080/tcp", "WEB :8000/tcp", and "WEBSECURE :8443/tcp". Below this is a "HTTP" section with three circular dashboards: "Routers", "Services", and "Middlewares". Each dashboard shows success rates and counts for Success, Warnings, and Errors.

Metric	Routers	Services	Middlewares
Success	100%	100%	0%
Warnings	0%	0%	0%
Errors	0%	0%	0%

Traefik Installation - Helm

- GatewayAPI CRD
- GatewayClass
- Gateway



```
$ helm upgrade --install --namespace traefik traefik  
traefik/traefik --create-namespace --set  
providers.kubernetesGateway.enabled=true
```

Traefik Installation - Helm

- GatewayAPI CRD
- GatewayClass
- Gateway opt-out



```
$ helm upgrade --install --namespace traefik traefik  
traefik/traefik --create-namespace --set  
providers.kubernetesGateway.enabled=true \  
--set gateway.enabled=false
```

Traefik Installation - Helm

- GatewayAPI CRD
- GatewayClass
- Gateway opt-out



```
$ helm upgrade --install --namespace traefik traefik  
traefik/traefik --create-namespace --set  
providers.kubernetesGateway.enabled=true \  
--set gateway.enabled=false
```

```
traefik with docker.io/traefik:v3.3.5 has been  
deployed successfully on traefik namespace !
```

```
$ kubectl -n traefik describe deployment traefik
```

```
...
```

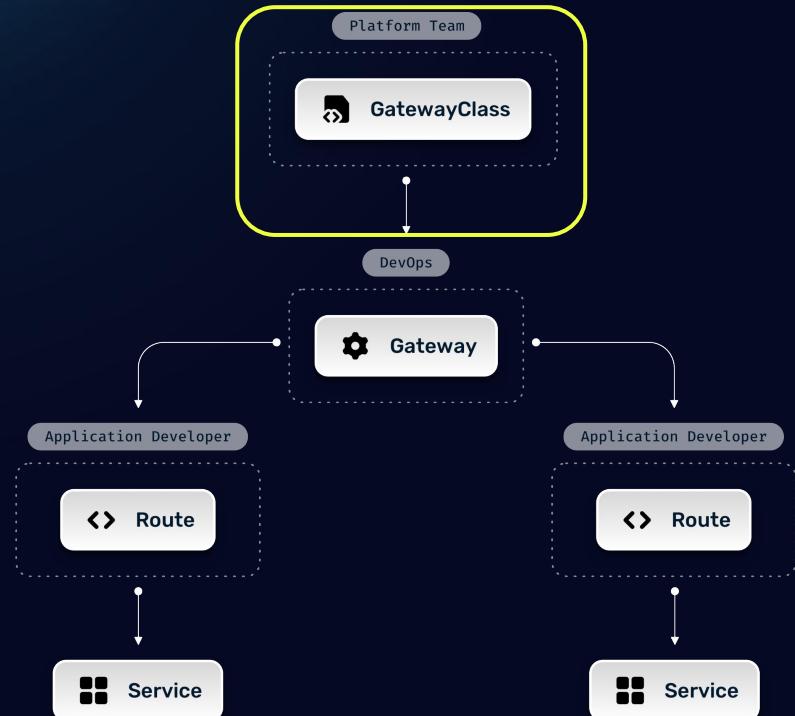
Args:

```
--entryPoints.web.address=:8000/tcp  
--entryPoints.websecure.address=:8443/tcp  
--entryPoints.websecure.http.tls=true  
--providers.kubernetesgateway
```

```
...
```

GatewayClass

- Similar to IngressClass
- Define Gateway Controller to use



GatewayClass

- Specifications:

- controllerName

Controller linked to the GatewayClass



```
$ kubectl get gatewayclasses.gateway.networking.k8s.io traefik -o yaml
```

```
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: traefik
spec:
  controllerName: traefik.io/gateway-controller
```

GatewayClass

- Specifications:

- controllerName
- parametersRef

Controller Configuration Resource



```
$ kubectl get gatewayclasses.gateway.networking.k8s.io traefik -o yaml
```

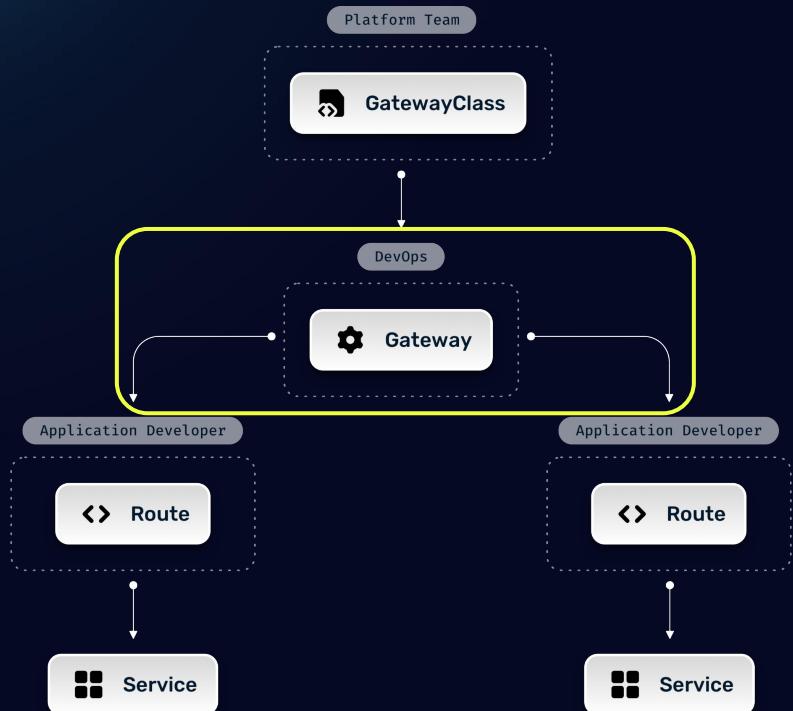
```
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: traefik
spec:
  controllerName: example.net/gateway-controller
  parametersRef:
    group: example.net
    kind: Config
    name: internet-gateway-config
```

DevOps Resources

Gateway, Listener

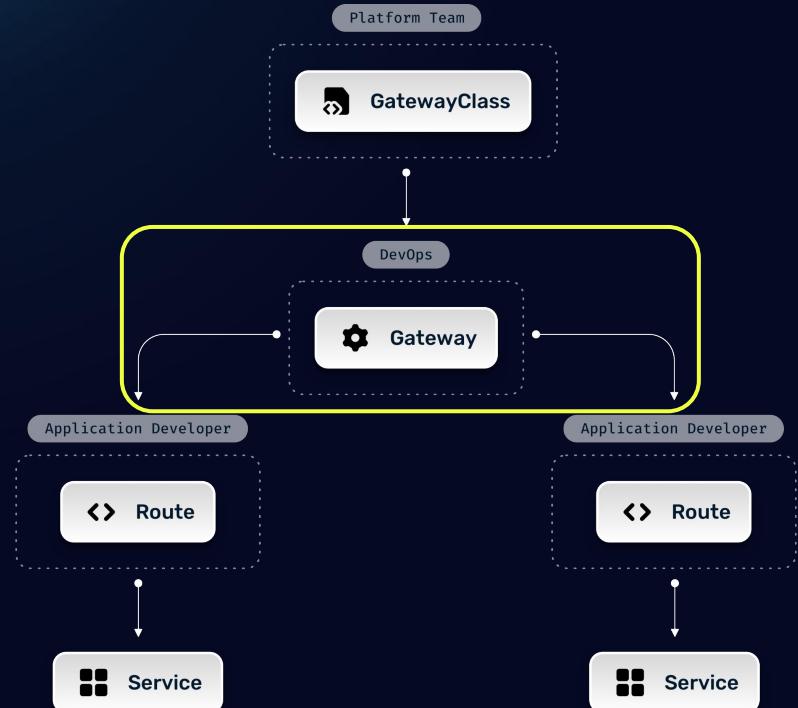
Gateway

- Entrypoints on the cluster



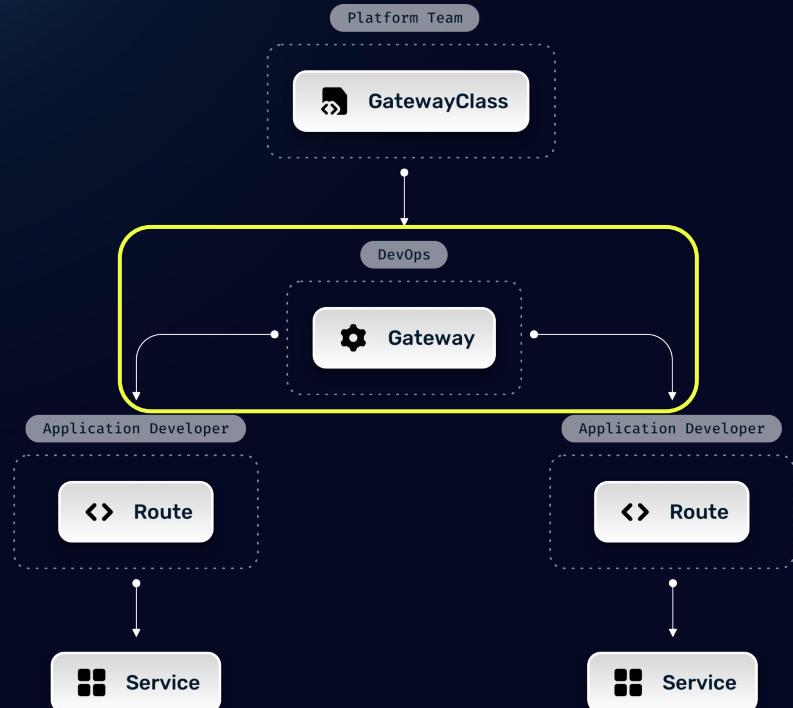
Gateway

- Entrypoints on the cluster
- Rules on routes allowed



Gateway

- Entrypoints on the cluster
- Rules on routes allowed
- Multi-tenancy



Gateway

- Specifications

- GatewayClassName

GatewayClass used for the Gateway



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
```

Gateway

- Specifications
 - gatewayClassName
 - listeners
 - Key Concept
 - EntryPoints definition
 - Route restriction rules



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: web
      port: 8000
      protocol: HTTP
      hostname: "example.com"
      allowedRoutes:
        namespaces:
          from: Same
```

Listener

- Specification

- port

Port to listen



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: web
      port: 8000
      protocol: HTTP
      hostname: "example.com"
      allowedRoutes:
        namespaces:
          from: Selector
          selector:
            matchLabels:
              access: public
      kinds:
        - kind: HTTPRoute
```

Listener

- Specification

- port
- protocol
 - Protocol expected
 - HTTP, HTTPS
 - TCP/TLS, UDP (experimental)

```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: web
      port: 8000
      protocol: HTTP
      hostname: "example.com"
      allowedRoutes:
        namespaces:
          from: Selector
          selector:
            matchLabels:
              access: public
      kinds:
        - kind: HTTPRoute
```

Listener

- Specification

- port
- protocol
- hostname

Request HostName / SNI



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: web
      port: 8000
      protocol: HTTP
      hostname: "example.com"
      allowedRoutes:
        namespaces:
          from: Selector
          selector:
            matchLabels:
              access: public
      kinds:
        - kind: HTTPRoute
```

Listener

- Specification

- port
- protocol
- hostname
- hostname, allowedRoutes

Restriction on the attached routes



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: web
      port: 8000
      protocol: HTTP
      hostname: "example.com"
      allowedRoutes:
        namespaces:
          from: Selector
          selector:
            matchLabels:
              access: public
        kinds:
          - kind: HTTPRoute
```

Listener - HTTPS / TLS

- Specification
 - `tls.mode`
 - Terminate

```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "secure.example.com"
      allowedRoutes:
        namespaces:
          from: Same
      tls:
        mode: Terminate
        certificateRefs:
          - kind: Secret
            name: example-cert
    - name: tls-passthrough
      port: 9444
      protocol: TLS
      hostname: "secure-sni.example.com"
      tls:
        mode: Passthrough
```

Listener - HTTPS / TLS

- Specification

- `tls.mode`
 - Terminate
 - Passthrough (only TLS)



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "secure.example.com"
      allowedRoutes:
        namespaces:
          from: Same
      tls:
        mode: Terminate
        certificateRefs:
          - kind: Secret
            name: example-cert
    - name: tls-passthrough
      port: 9444
      protocol: TLS
      hostname: "secure-sni.example.com"
      tls:
        mode: Passthrough
```

Listener - HTTPS / TLS

- Specification

- `tls.mode`
- `tls.certificateRefs`

Only for TLS termination



```
# 01-gateway.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: traefik-gateway
  namespace: traefik
spec:
  gatewayClassName: traefik
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "secure.example.com"
      allowedRoutes:
        namespaces:
          from: Same
      tls:
        mode: Terminate
        certificateRefs:
        - kind: Secret
          name: example-cert
    - name: tls-passthrough
      port: 9444
      protocol: TLS
      hostname: "secure-sni.example.com"
      tls:
        mode: Passthrough
```

Dev Resources

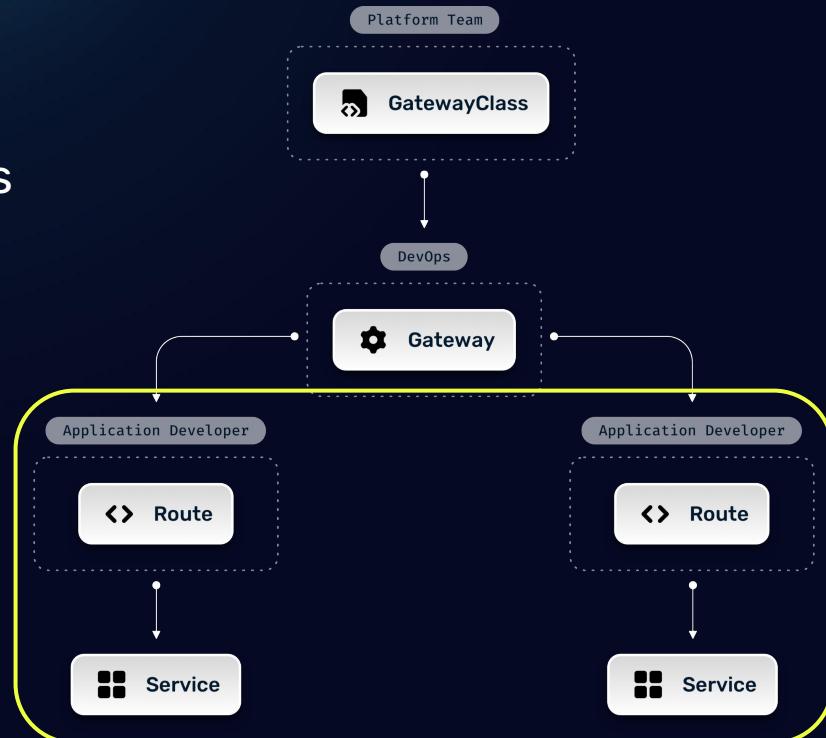
Routes, Filters

Routes

- Routing rules to Services
- Operations on the requests/responses

Routes

- Routing rules to Services
- Operations on the requests/responses
- Protocol oriented
 - `HTTPRoute`, `GRPCRoute`
 - `TCPRoute/TLSRoute` (experimental)
 - `UDPRoute` (experimental)



HTTPRoute

- Specification

- parentsRef

Gateways to expose the Route



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: who
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
```

HTTPRoute

- Specification

- parentsRef
- hostnames

List of hostnames to match



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: who
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
    - "*.example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
```

HTTPRoute

- Specification

- parentsRef
- hostnames
- rules.matches

Request Matchers
(path, header, method, queryParam)



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: who
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: PathPrefix
            value: /api
        method: GET
        headers:
          - name: x-app-version
            value: "v2"
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
```

HTTPRoute

- Specification

- parentsRef
- hostnames
- rules.matches
- rules.backendRefs

Backends to reach



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: who
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: main-backend
      port: 8080
      weight: 80
    - name: canary-backend
      port: 8080
      weight: 20
```

HTTPRoute

- Specification

- parentsRef
- hostnames
- rules.matches
- rules.backendRefs
- rules.filters

Operations on requests/responses



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: who
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
  filters:
    - type: RequestHeaderModifier
      requestHeaderModifier:
        add:
          - name: x-added-header
            value: added-by-gateway
```

HTTP Filters

- **Core Filters**

Filters the Controllers MUST provide



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: traefik
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
  filters:
    - type: ResponseHeaderModifier
      responseHeaderModifier:
        add:
          - name: X-Header-Add-1
            value: header-add-1
```

HTTP Filters

- **Core Filters**

Filters the Controllers MUST provide

- **Extended Filters**

Filters the Controllers CAN provide



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: traefik
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
  filters:
    - type: URLRewrite
      urlRewrite:
        hostname: "backend.svc.cluster.local"
```

HTTP Filters

- **Core Filters**

Filters the Controllers MUST provide

- **Extended Filters**

Filters the Controllers CAN provide

- **ExtensionRef Filters**

Additional filters



```
# 02-httproute.yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: who-httproute
  namespace: traefik
spec:
  parentRefs:
    - name: traefik-gateway
      kind: Gateway
      namespace: traefik
      sectionName: websecure
  hostnames:
    - "example.com"
  rules:
    - matches:
        - path:
            type: Exact
            value: /healthz
  backendRefs:
    - name: my-backend
      port: 8080
  filters:
    - type: ExtensionRef
      extensionRef:
        group: traefik.io
        kind: Middleware
        name: rate-limit
```

To sum-up



Expose your first HTTPRoute!

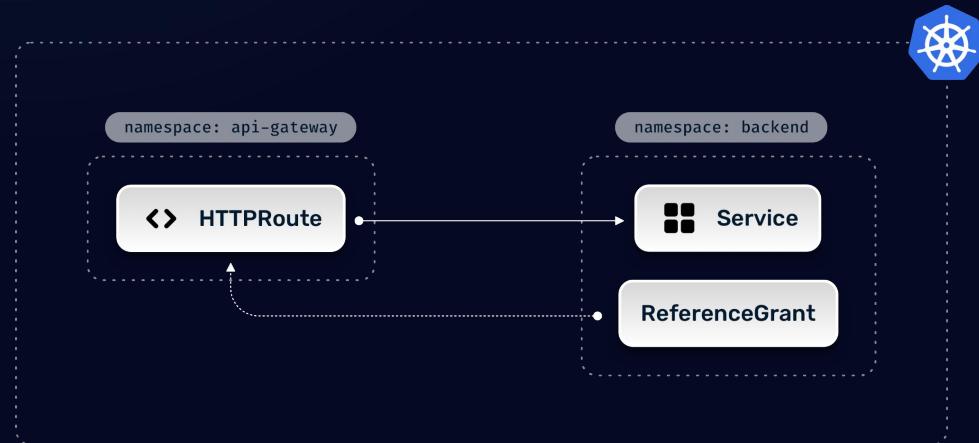


Let's dig deeper!



ReferenceGrant

- Cross Namespace Reference 🎉



ReferenceGrant

- **Cross Namespace Reference** 🎉
- Route -> Service



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: allow-prod-traffic
  namespace: targeted-ns
spec:
  from:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    namespace: prod
  to:
  - kind: Service
    group: ""
```

ReferenceGrant

- **Cross Namespace Reference** 🎉
- Route -> Service
- Gateway -> TLS Secret



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: tls-termination
  namespace: route-ns
spec:
  from:
  - group: gateway.networking.k8s.io
    kind: Gateway
    namespace: traefik
  to:
  - kind: Secret
    group: ""
```

ReferenceGrant

- **Cross Namespace Reference** 🎉
- Route -> Service
- Gateway -> TLS Secret

⚠️ Gateway -> Route cross namespace
using allowedRoutes



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: tls-termination
  namespace: route-ns
spec:
  from:
    - group: gateway.networking.k8s.io
      kind: Gateway
      namespace: traefik
  to:
    - kind: Secret
      group: ""
```

ReferenceGrant

- Deployed in the child object namespace



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: allow-prod-traffic
  namespace: targeted-ns
spec:
  from:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    namespace: prod
  to:
  - kind: Service
    group: ""
```

ReferenceGrant

- Deployed in the child object namespace
- Specifications
 - From
Kind of parent object + namespace



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: allow-prod-traffic
  namespace: targeted-ns
spec:
  from:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    namespace: prod
  to:
  - kind: Service
    group: ""
```

ReferenceGrant

- Deployed in the targeted namespace
- Specifications
 - From
 - To

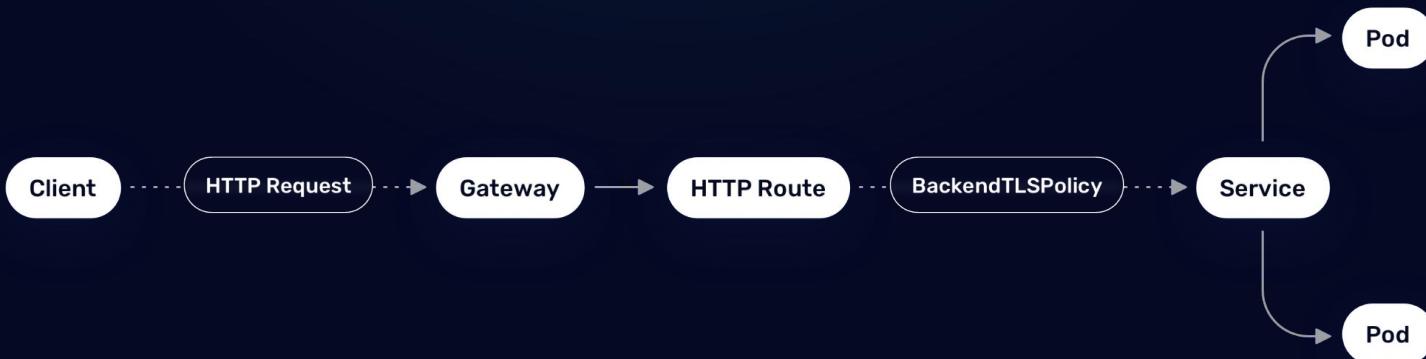
Kind of child object (+ name)



```
# 03-ref-grant.yaml
apiVersion: gateway.networking.k8s.io/v1beta1
kind: ReferenceGrant
metadata:
  name: allow-prod-traffic
  namespace: targeted-ns
spec:
  from:
  - group: gateway.networking.k8s.io
    kind: HTTPRoute
    namespace: prod
  to:
  - kind: Service
    group: ""
    name: my-svc
```

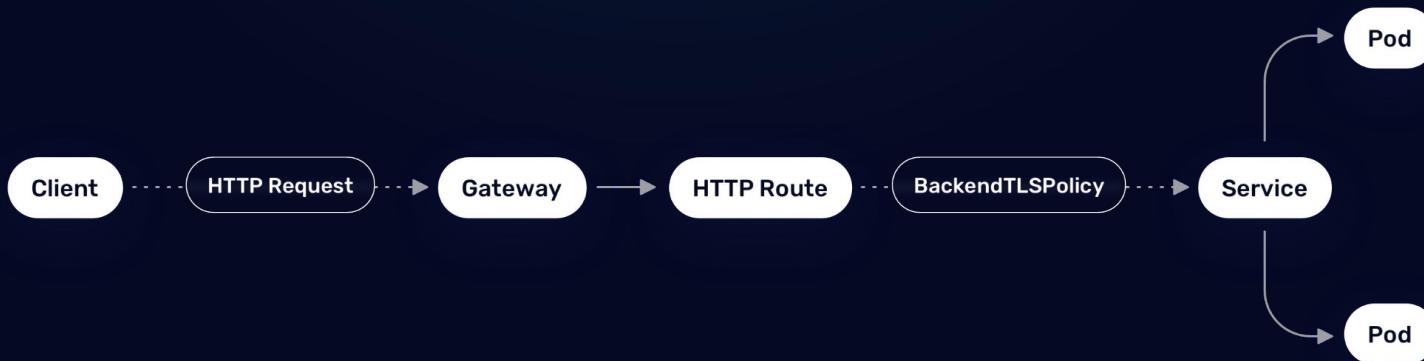
BackendTLSPolicy

- **TLS Termination on the backend...**



BackendTLSPolicy

- **TLS Termination on the backend... Keeping HTTP capabilities!**



BackendTLSPolicy

- Specifications

- validation.hostname

Request SNI



```
# 04-backend-tls-policy.yaml
kind: BackendTLSPolicy
apiVersion: gateway.networking.k8s.io/v1alpha3
metadata:
  name: whoami-policy
  namespace: traefik
spec:
  validation:
    hostname: whoami.docker.localhost
  caCertificateRefs:
    - group: ""
      kind: ConfigMap
      name: internal-ca
  targetRefs:
    - group: ""
      kind: Service
      name: whoami-tls
```

BackendTLSPolicy

- Specifications

- validation.hostname
- validation.caCertificateRefs

CA Root backend certificates



```
# 04-backend-tls-policy.yaml
kind: BackendTLSPolicy
apiVersion: gateway.networking.k8s.io/v1alpha3
metadata:
  name: whoami-policy
  namespace: traefik
spec:
  validation:
    hostname: whoami.docker.localhost
    caCertificateRefs:
      - group: ""
        kind: ConfigMap
        name: internal-ca
targetRefs:
  - group: ""
    kind: Service
    name: whoami-tls
```

BackendTLSPolicy

- Specifications

- validation.hostname
- validation.caCertificateRefs
- targetRefs

Services/Pods to connect to



```
# 04-backend-tls-policy.yaml
kind: BackendTLSPolicy
apiVersion: gateway.networking.k8s.io/v1alpha3
metadata:
  name: whoami-policy
  namespace: traefik
spec:
  validation:
    hostname: whoami.docker.localhost
    caCertificateRefs:
      - group: ""
        kind: ConfigMap
        name: internal-ca
  targetRefs:
    - group: ""
      kind: Service
      name: whoami-tls
```

Everything together!

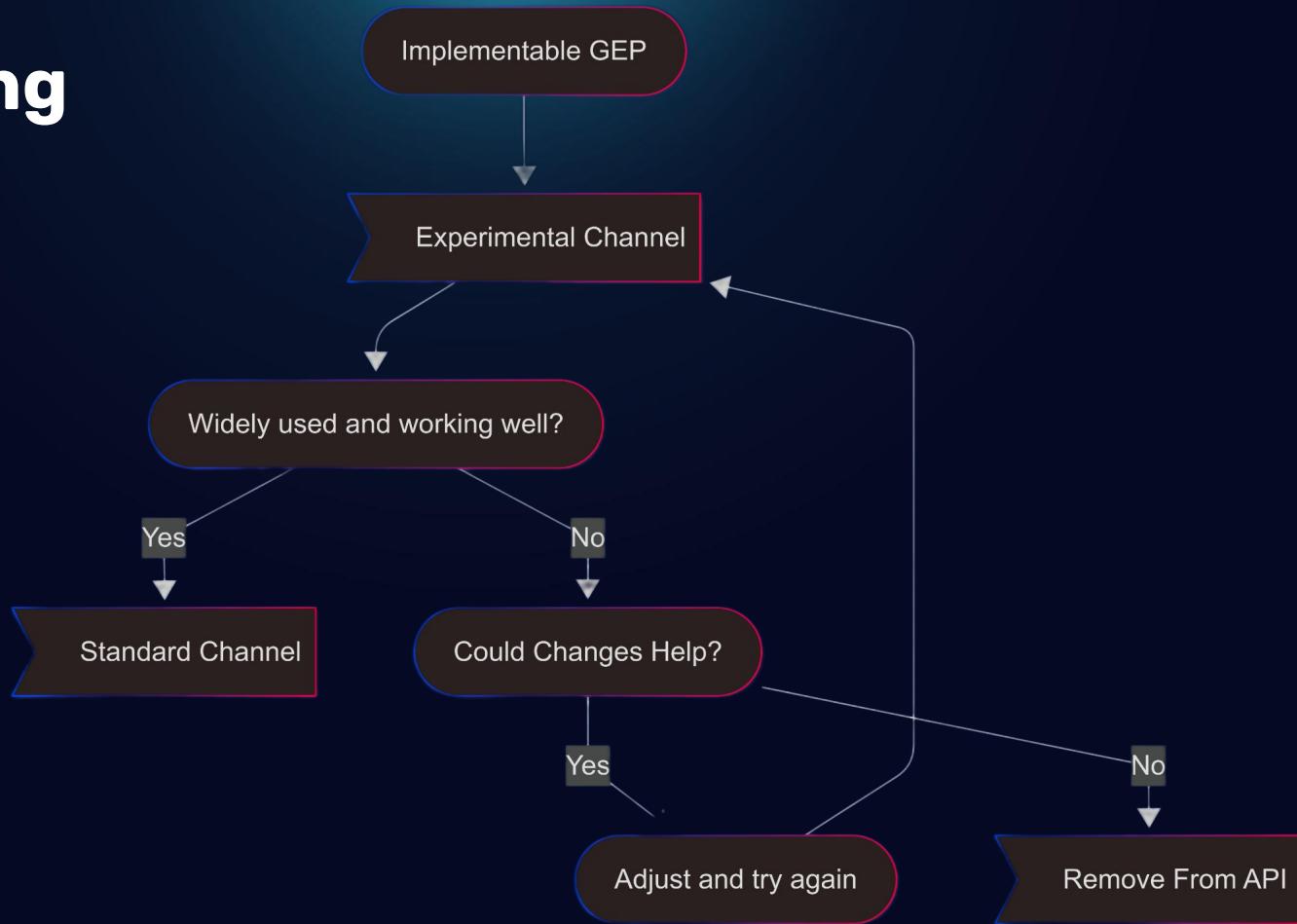


What's Next?



Versioning

Gateway Enhancement Proposal



Current Limitations

- mTLS w/ backend servers: [GEP-3155](#)
- Stickiness: [GEP-1619](#)
- CORS filter: [GEP 1767](#)
- BackendTLSPolicy: [GEP-1897](#)
- Granular observability

More [experiments](#)



v1.3.0 RC1

- CORS filters, XListenerSets (Standard Mechanism to Merge Gateways), and XBackendTrafficPolicy (Retry Budgets) → Experimental Channel
- Percentage-based request mirroring → Standard Channel
- Improvements to OverlappingTLSConfig & BackendTLSPolicy
- Upgrade to Kubernetes v1.32 and Go v1.24
- Documentation improvements

→ [Full Changelog](#)



Thank you!



emilevauge



nmengin

