



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 Development*

What is Traefik?

A cloud-native & lightweight gateway to expose all your services



Traffic Management



Security



Observability

The screenshot shows the Traefik 3.4.0 dashboard interface. On the left is a sidebar with navigation links: Dashboard, HTTP (HTTP Routers, HTTP Services, HTTP Middlewares), TCP (TCP Routers, TCP Services, TCP Middlewares), UDP (UDP Routers, UDP Services), Plugins, and Features (Tracing, Metrics, AccessLog). The main area is divided into sections: Entrypoints (listing TRAEFIK :8080, WEB :8000, WEB-SSL :8443, and WEB-REDIRECT :80), HTTP (Routers, Services, Middlewares), TCP (Routers, Services, Middlewares), UDP (Routers, Services, Middlewares), and Features (Metrics set to OFF, AccessLog set to ON). Providers listed at the bottom include Docker, File, Marathon, KubernetesIngress, and KubernetesCRD.

- Entrypoints**
 - TRAEFIK :8080
 - WEB :8000
 - WEB-SSL :8443
 - WEB-REDIRECT :80
- HTTP**
 - Routers**: Success 42, Warnings 42, Errors 42
 - Services**: Success 83, Warnings 38, Errors 5
 - Middlewares**: Success 101, Warnings 10, Errors 15
- TCP**
 - Routers**: Success 74, Warnings 20, Errors 32
 - Services**: Success 111, Warnings 8, Errors 7
 - Middlewares**: Success 92, Warnings 23, Errors 11
- UDP**
 - No related objects to show.
- Features**
 - TRACING: Prometheus
 - METRICS: OFF
 - ACCESSLOG: ON
- Providers**
 - Docker
 - File
 - Marathon
 - KubernetesIngress
 - KubernetesCRD

900+

Contributors

60K+

Stars on Github

3.4B+

Downloads



GitOps-Driven, Scalable Solutions

1. Application Proxy

Simplify and automate the discovery, routing, and load balancing of microservices and virtual machines.



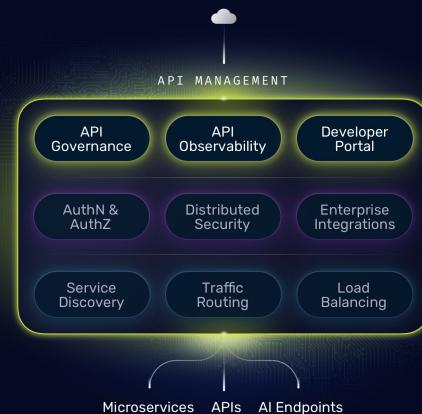
2. API / AI / MCP Gateway

When security and centralized control become priorities, teams upgrade to an API gateway.



3. API Management

As API lifecycle management becomes a priority, teams need additional capabilities beyond what API gateways can provide.



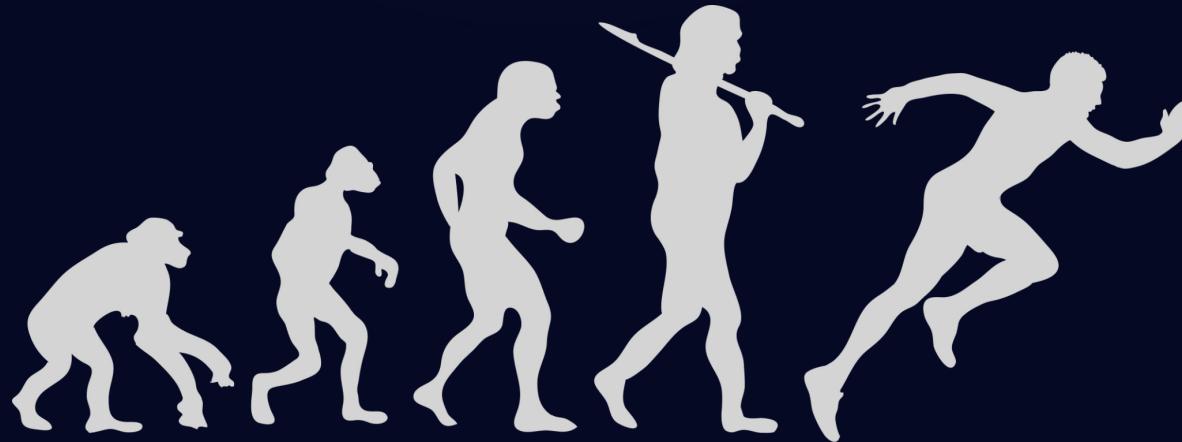
Why a new specification?

Ingress
(2015)

Vendor Neutral-*ish*
Specific Configuration Using Annotations
Only HTTP
Deprecated

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
```

100+



Ingress ❤️ Simplicity 💔 Reality

😊 Simple

- 😢 Limited expressiveness
- 😢 No role separation
- 😢 Single tenant
- 😢 HTTP only
- 😢 Single resource, multiple concerns
- 😢 No status feedback

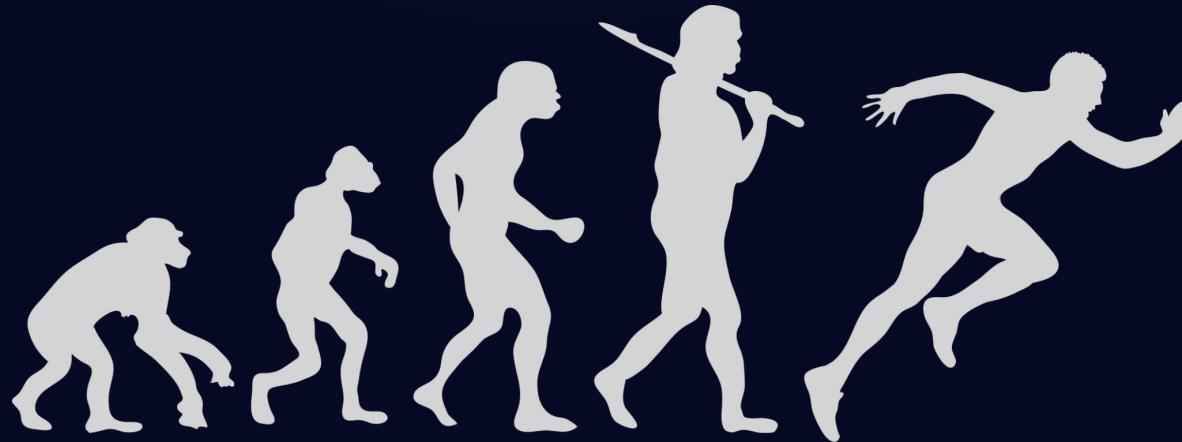
Why a new specification?

Ingress
(2015)

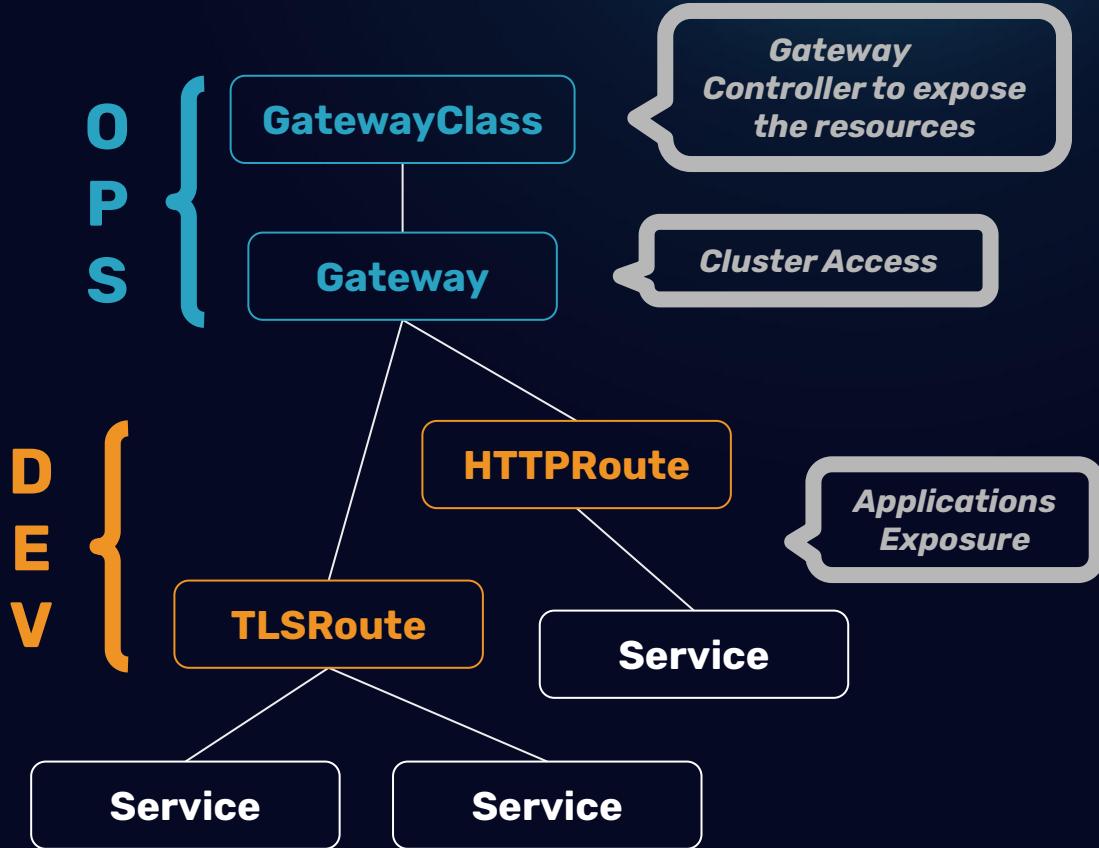
Vendor Neutral-*ish*
Specific Configuration Using Annotations
Only HTTP
Deprecated

Gateway API
(2019)

Vendor Neutral (**Expected**)
Configuration Using Objects
HTTP/TCP
Next Standard
Active Roadmap
Ingress and Mesh



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 into 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 v1.4.0 Traefik Proxy

Gateway API Conformance v1.2.1 Kong Ingress Controller



Gateway Controller

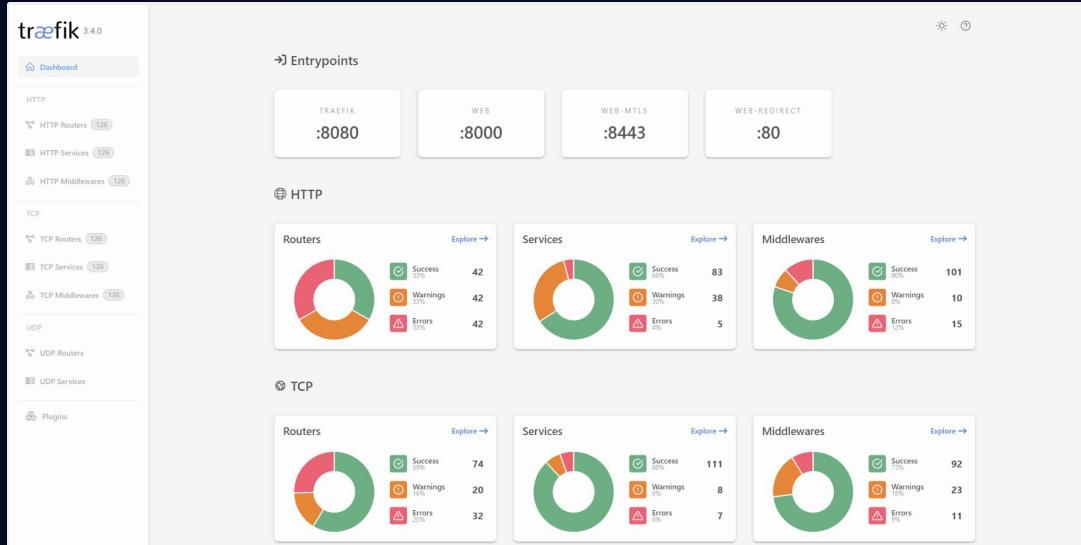
- Many Implementations
- Support Levels
- Conformance Tests
- Choose the best for you!





A lightweight, Gateway Controller.

Open Source • Traffic Management • Security • Observability



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

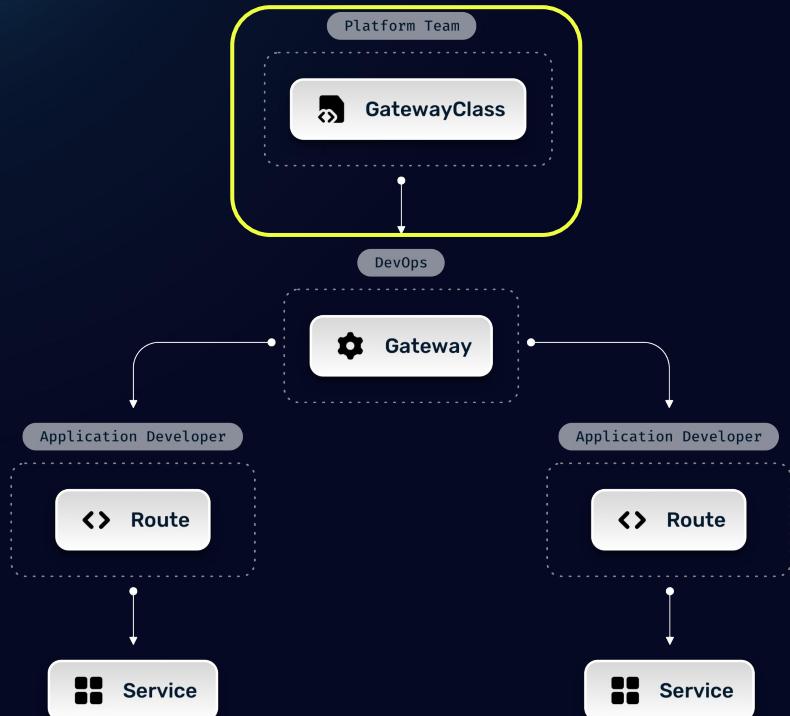
- GatewayAPI CRD
- GatewayClass
- Gateway opt-out



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

GatewayClass

- Similar to IngressClass
- Define Gateway Controller to use



GatewayClass

- Specification:

- controllerName

Controller linked to the GatewayClass



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

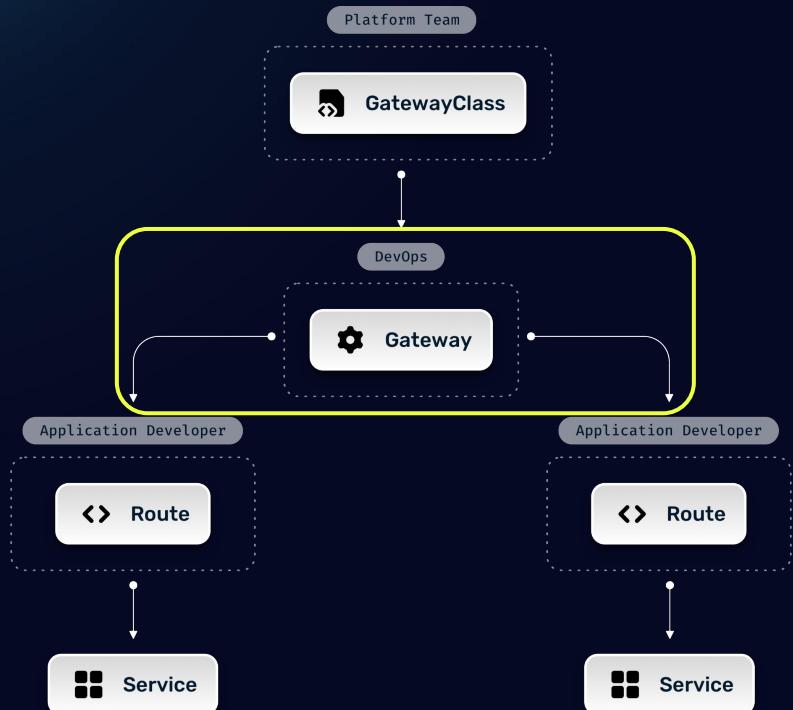
```
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: traefik
spec:
  controllerName: traefik.io/gateway-controller
```

DevOps Resources

Gateway, Listener

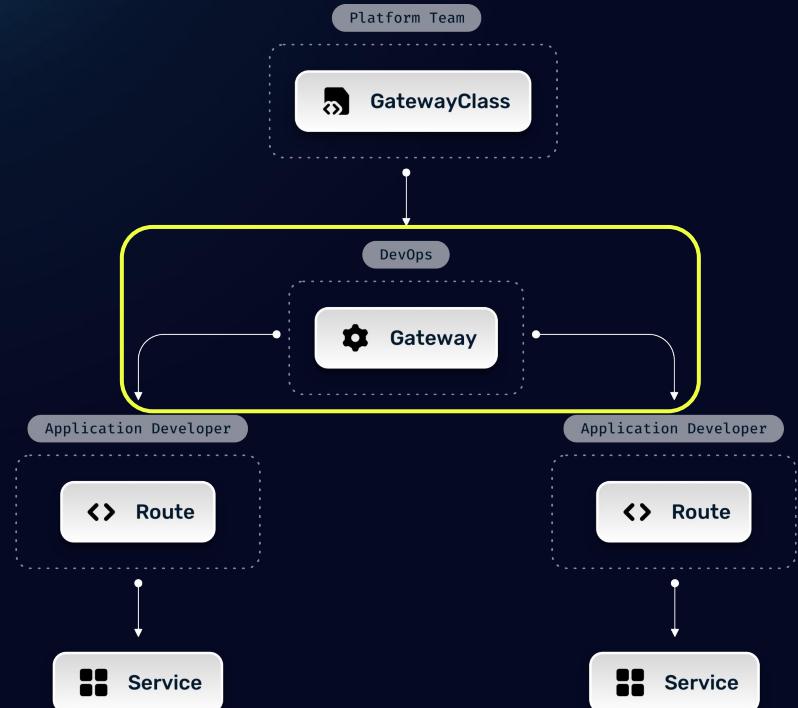
Gateway

- Entrypoints on the cluster



Gateway

- Entrypoints on the cluster
- Rules on routes allowed



Gateway

- Specification

- 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

- Specification
 - 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

```
# 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
- 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

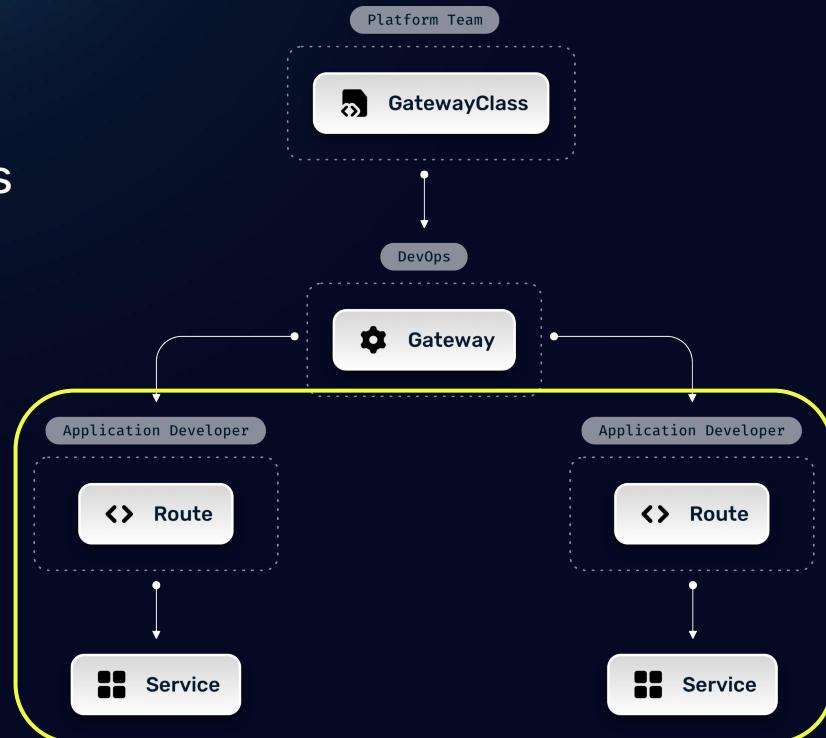
Routes, Filters

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

- parentsRefs

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

- parentsRefs
- 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

- parentsRefs
- 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

- parentsRefs
- 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

- parentsRefs
- 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

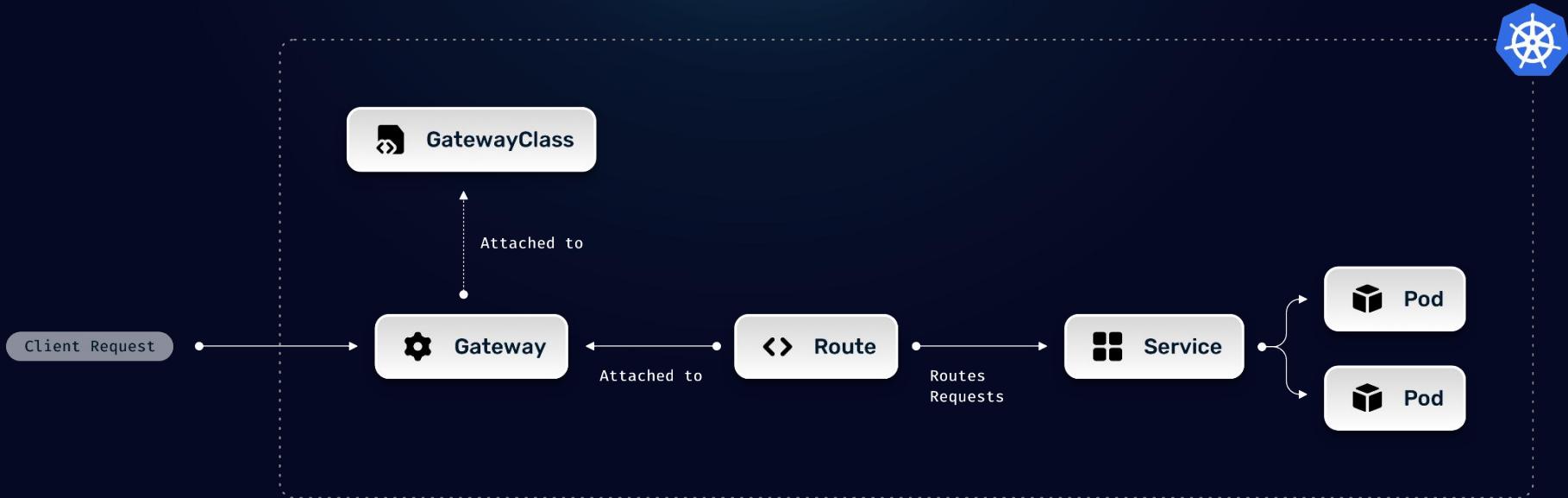
- **ExtensionRef Filters**

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



Let's expose a first HTTPRoute!



treefik_values.yaml

```
traefik_values.yaml  00-whoami.yaml  01-gateway.yaml  02-HTTPRoute.yaml
```

1 image:
2 pullPolicy: Always
3
4 # Enable the dashboard and exposes it on the entrypoint web (port :80)
5 ingressRoute:
6 dashboard:
7 enabled: true
8 matchRule: "Host(`dashboard.docker.localhost`)"
9 entryPoints:
10 - "web"
11
12 providers:
13 # Disable Ingress provider... we do not need it anymore ;-)
14 kubernetesIngress:
15 enabled: false
16 # Enable the GatewayAPI provider
17 # A GatewayClass will be created too
18 kubernetesGateway:
19 enabled: true
20
21 gateway:
22 # Avoid deploying a default gateway
23 enabled: false

gateway-api-demo> k3d cluster create demo --port 80:80@loadbalancer --port 443:443@loadbalancer --k3s-arg "--disable=traefik@server:0"
INFO[0000] portmapping '80:80' targets the loadbalancer: defaulting to [servers*:*:proxy agents*:*:proxy]
INFO[0000] portmapping '443:443' targets the loadbalancer: defaulting to [servers*:*:proxy agents*:*:proxy]
INFO[0000] Prep: Network
INFO[0000] Created network 'k3d-demo'
INFO[0000] Created image volume K3d-demo-images
INFO[0000] Starting new tools node...
INFO[0000] Starting node 'k3d-demo-tools'
INFO[0001] Creating node 'k3d-demo-server-0'
INFO[0001] Creating LoadBalancer 'k3d-demo-serverlb'
INFO[0001] Using the k3d-tools node to gather environment information
INFO[0001] HostIP: using network gateway 192.168.97.1 address
INFO[0001] Starting cluster 'demo'
INFO[0001] Starting servers...
INFO[0001] Starting node 'k3d-demo-server-0'
INFO[0003] All agents already running.
INFO[0003] Starting helpers...
INFO[0003] Starting node 'k3d-demo-serverlb'

What's Next?



Still Experimental

- mTLS w/ backend servers: [GEP-3155](#)
- Stickiness: [GEP-1619](#)
- Retry Budgets: [GEP-3388](#)
- Granular observability

More [experiments](#)



v1.5.0 RC1

- **Standard**
 - TLSRoute
 - CORS
 - ListenerSets
- **Experimental**
 - HTTP Authentication
 - Default Gateway



And This Happened...

The Dark Side of Open Source

Lack of Resources



Small, Dedicated Team

For years, the project relied on just 1–2 core maintainers working largely in their spare time.



Overwhelming Demand

The volume of issues, feature requests, and security triage required for such a critical component far exceeded the capacity of a volunteer team.

IngressNightmare



CVE-2025-1974—March 24, 2025



**9.8 Critical Unauthenticated
Remote Code Execution
Vulnerabilities in Ingress
NGINX**



**Over 40% of cloud
environments are vulnerable
to RCE**

Unauthenticated admission-controller flaws allow attackers to inject malicious NGINX config via Ingress annotations or UID. Combined with a shared-library upload via client-body buffering, this yields Remote Code Execution, exposing cluster-wide secrets and enabling full Kubernetes takeover

The InGate Failure

The "InGate" project (an attempted successor) failed to attract enough contributors to be viable, proving that the community could not support a new iteration either.

InGate

Caution

InGate is being retired (early 2026). SIG Network and the Security Response Committee recommend that all users begin migration to Gateway API or another Ingress controller immediately. Many options are listed in the Kubernetes documentation: [Gateway API](#) and [Ingress](#). Additional options may be available from vendors you work with.

An Inevitable Outcome

- ✓ Lack of Resources
- ✓ Unmanageable **Security Risks**
- ✓ InGate Failure



**Ingress NGINX
Retirement in
March 2026**

NGINX Ingress Provider #11844

Merged

traefiker merged 10 commits into `traefik:master` from `rtribotte:ingress-nginx`  yesterday

Conversation 21

Commits 10

Checks 51

Files changed 32



rtribotte commented 5 days ago • edited by nmengin ▾

Member ...

What does this PR do?

This PR introduces a new experimental provider to support Ingresses with NGINX annotations.

Motivation

To bootstrap and provide a solution for migrating from Ingresses with NGINX annotations.

Traefik ❤️ ingress-nginx

- ✓ Drop-in Nginx replacement foundation
- ✓ Secure-by-design
- ✓ Wide adoption & production proven
- ✓ Gateway API ready

**It is now possible to migrate from ingress-nginx at no cost,
while still being future proof and Gateway API ready.**

MIGRATION GUIDE

Ingress NGINX End of Life

Your Comprehensive
Migration Guide for Ingress
NGINX Controller
Retirement



49

DAYS

07

HOURS

18

MINUTES

04

SECONDS



Critical Notice

Ingress NGINX Controller **will be retired** in **March 2026** with no further security updates. Plan your migration strategy before hitting the deadline.

→ ingressnginxmigration.org



FEB 11–12, 2026

CONTAINER days LONDON

From POC to Production: Gateway API in Action



Emile Vauge Nicolas Mengin





Thank you!

 emilevauge

 nmengin

 træfiklabs

