



From POC to Production Gateway API in Action

Emile Vauge

Traefik *Creator*
Traefik Labs *CTO & Founder*



Nicolas Mengin

Traefik *Maintainer*
Traefik Labs *Head of Development*

Why a new Specification?



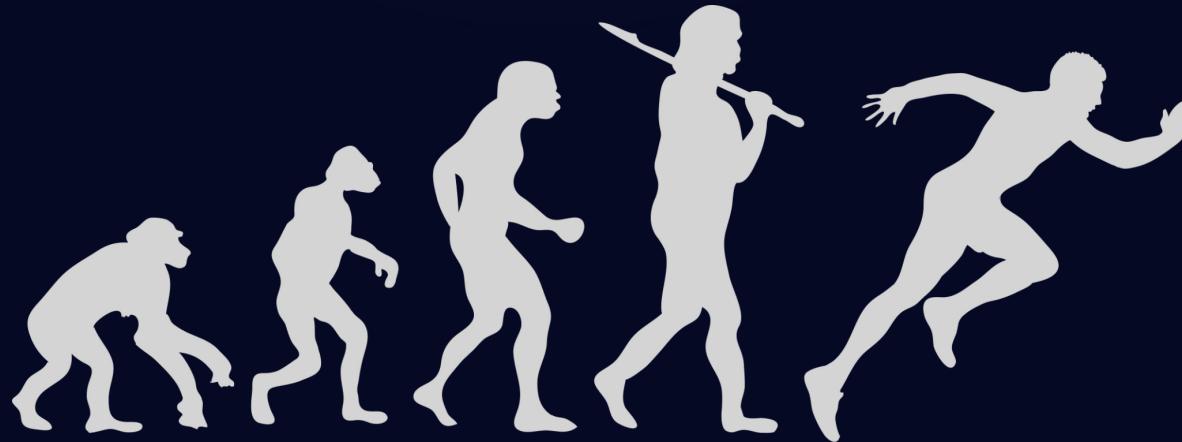
Why a new specification?

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 💔 Reality

😊 Simple

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

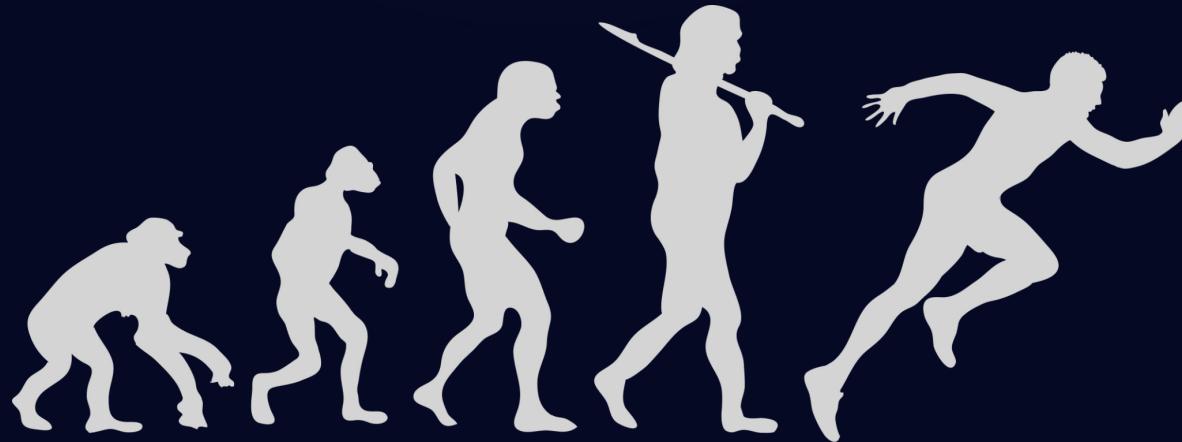
Why a new specification?

Ingress
(2015)

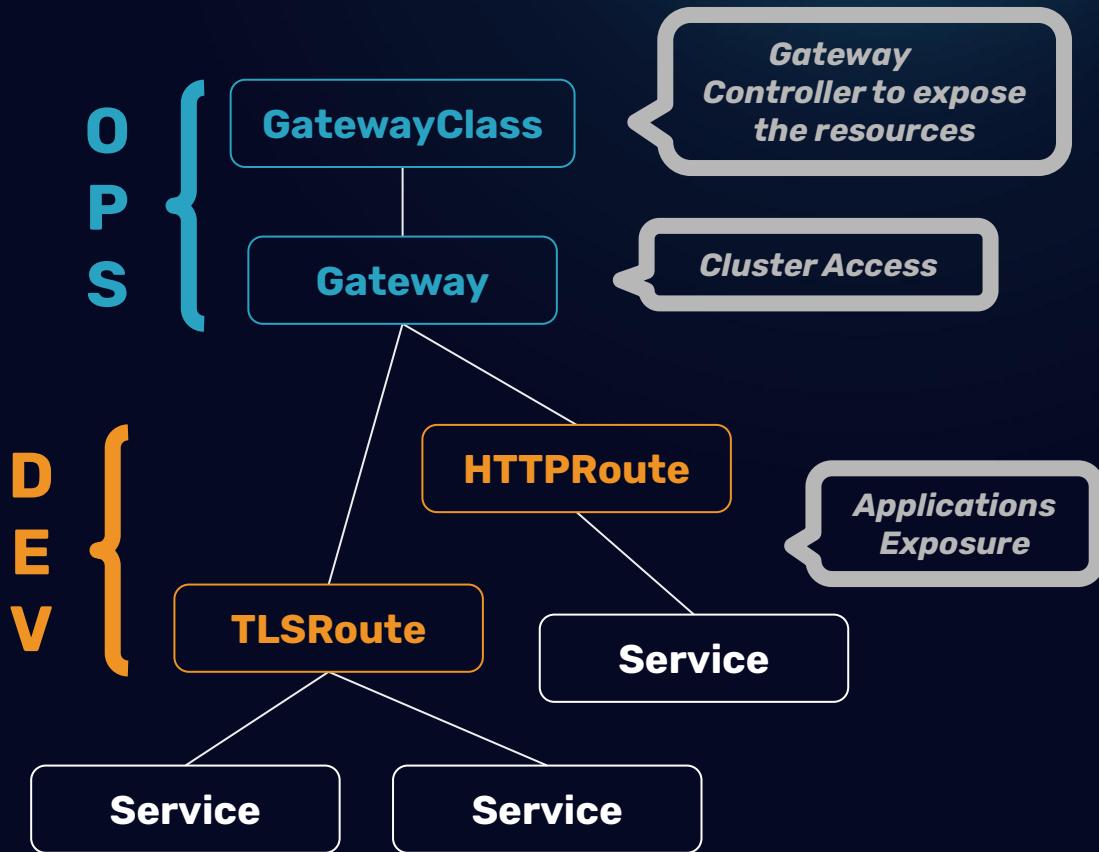
Vendor Neutral-*ish*
Specific Configuration Using Annotations
Only HTTP
Deprecated

Gateway API
(2019)

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



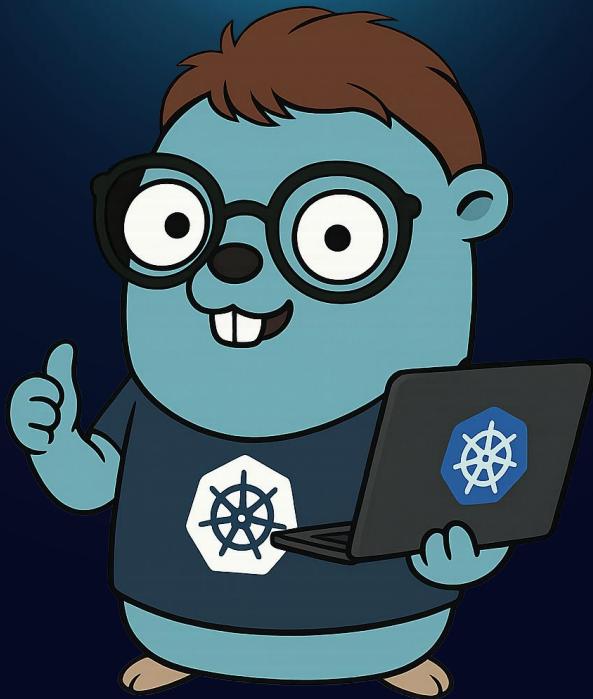
Core Objects



```
# 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:
        ...
```



GOPHY
DEVOPS

Production Ready?

- Core Objects = Ingress
 - Fit simple Use Cases



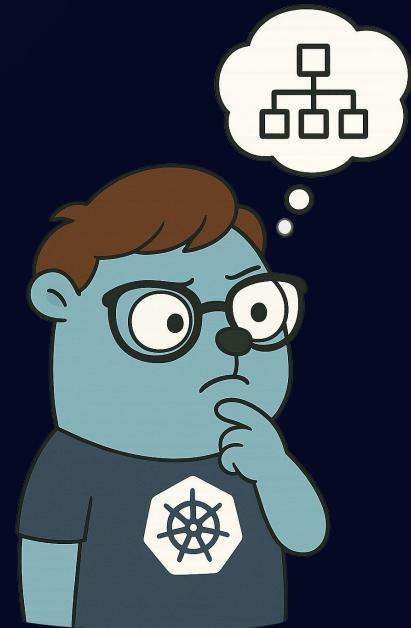
Production Ready?

- Core Objects = Ingress
 - Fit simple Use Cases
- What about Enterprise needs?



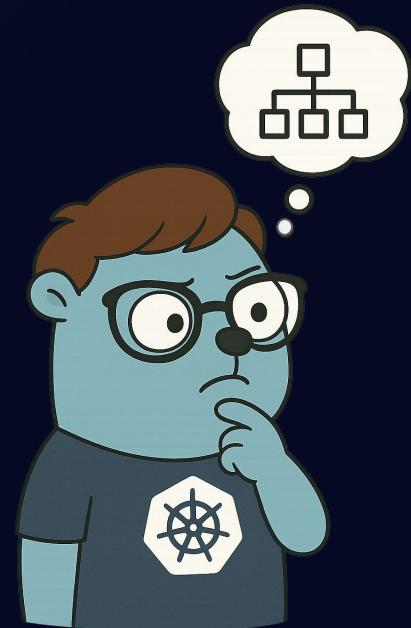
Production Ready?

- Core Objects = Ingress
 - Fit simple Use Cases
- What about Enterprise needs?
 - Extensible Specification!



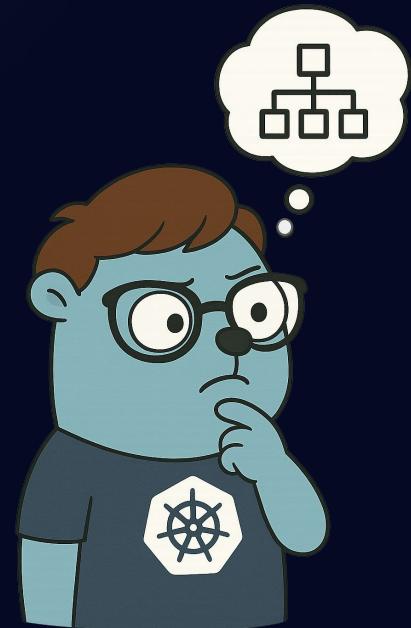
Production Ready?

- Core Objects = Ingress
 - Fit simple Use Cases
- What about Enterprise needs?
 - Extensible Specification!
- Let's solve two challenges!



Production Ready?

- Core Objects = Ingress
 - Fit simple Use Cases
- What about Enterprise needs?
 - Extensible Specification!
- Let's solve two challenges!
 - Security
 - Multi-Tenancy

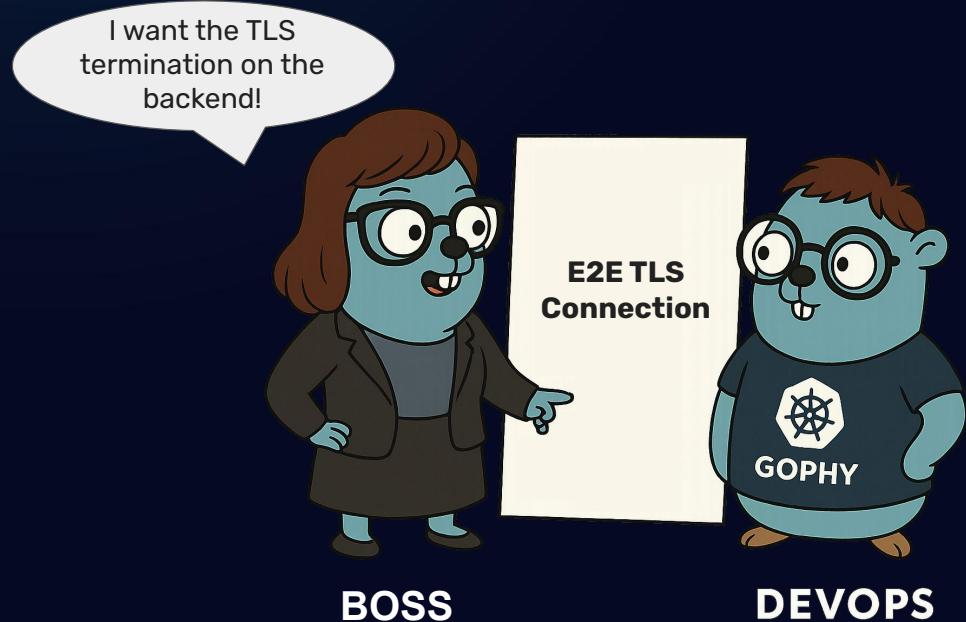


Security

End To End TLS Connection

- **Why a E2E Connection?**

- Zero Trust Security
- HTTP2
- gRPC
- etc



End To End TLS Connection

- **Challenge**
 - TLS Termination on the backend

End To End TLS Connection

- **Challenge**
 - TLS Termination on the backend
- **Solution**
 - TLSRoute

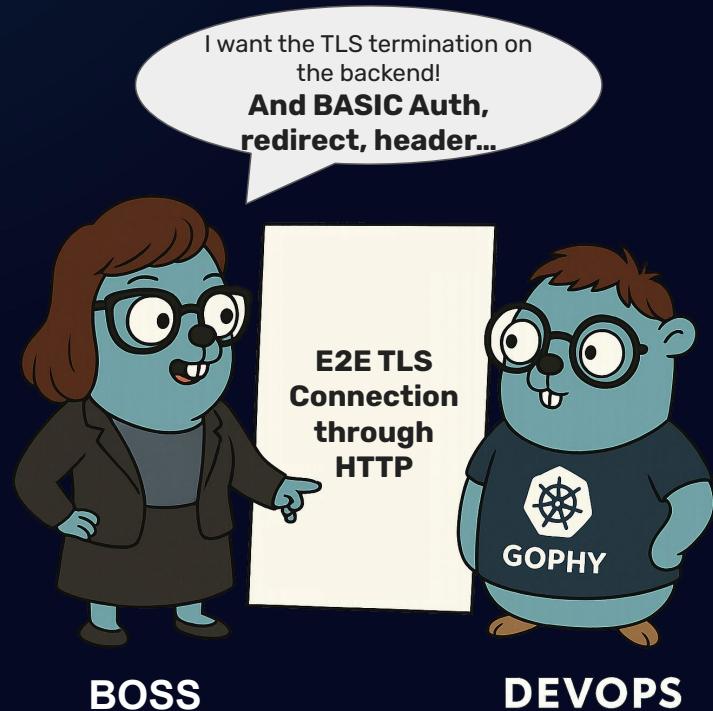
Demo

End To End TLS Connection

- **Challenge**
 - TLS Termination on the backend
- **Solution**
 - TLSRoute

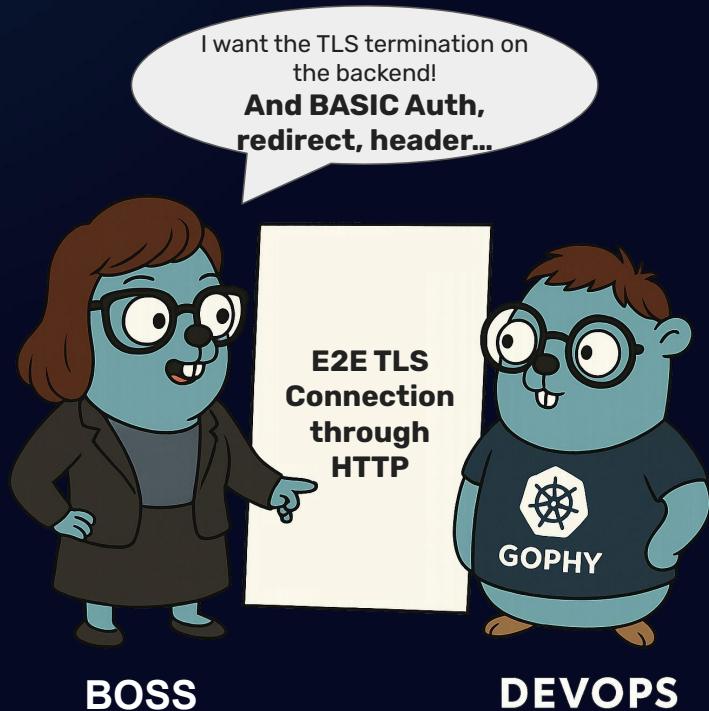
End To End TLS Connection

- **Challenge**
 - TLS Termination on the backend
- **Solution**
 - TLSRoute
- **But...**



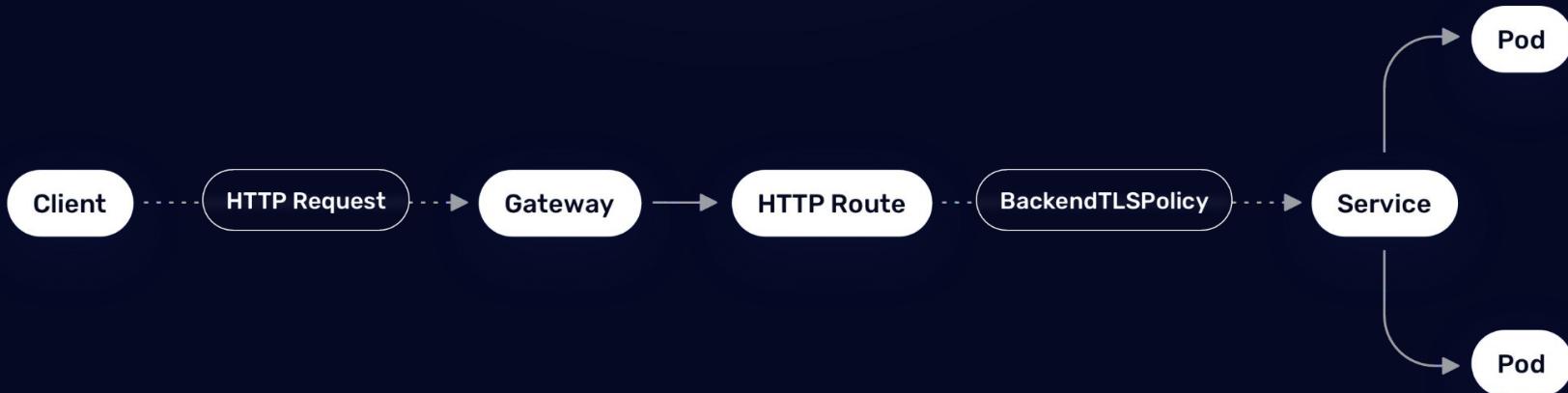
End To End TLS Connection

- **Challenge**
 - TLS Termination on the backend
- **Solution**
 - TLSRoute
- **But...**
 - TLSRoute = TCP
 - No HTTP Capabilities!



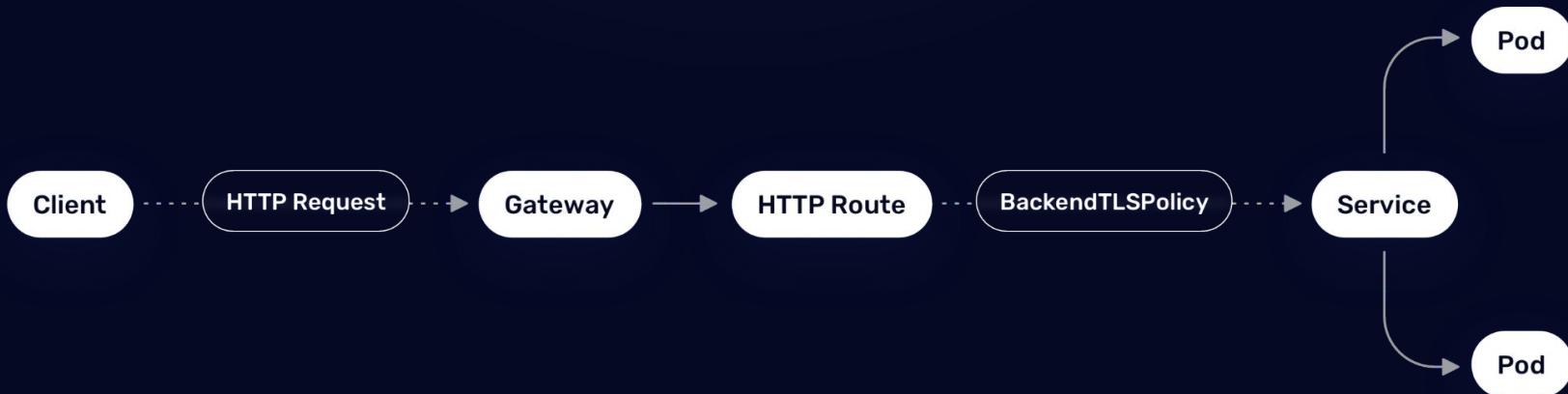
BackendTLSPolicy

- **How it works?**
 - TLS Termination on the Gateway



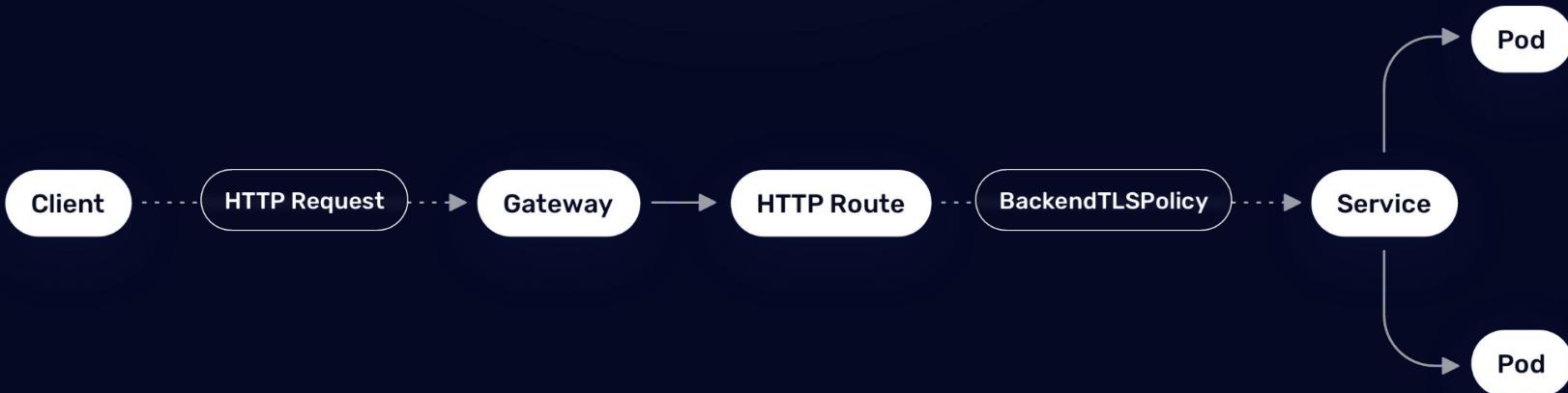
BackendTLSPolicy

- **How it works?**
 - TLS Termination on the Gateway
 - Backend HTTPS connection



BackendTLSPolicy

- **How it works?**
 - TLS Termination on the Gateway
 - Backend HTTPS connection
 - TLS Termination on the Backend

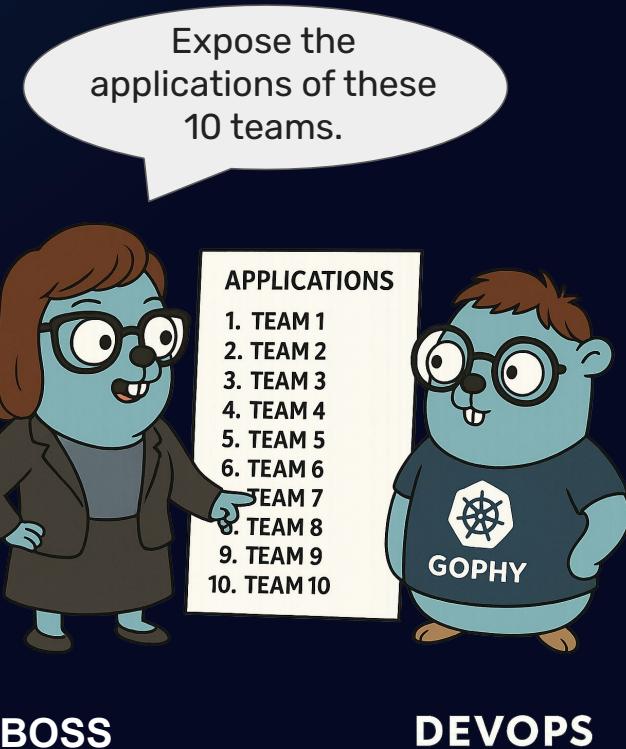


Demo

Multi-Tenancy

Multi-Tenancy

- **Why a multi-tenant cluster?**
 - Strong isolation boundaries
 - Noisy-neighbor prevention
 - Controlled access & policy enforcement
 - Failure containment



Multi-Tenancy

- **Challenges**
 - Let's the team managing the applications access...

Multi-Tenancy

- **Challenges**
 - Let's the team managing the applications access...
 - ... keeping the control of the cluster

Challenges

- **Challenges**
 - Let's the team managing the applications access...
 - ... keeping the control of the cluster
- **Solutions**
 - One namespace per team

Challenges

- **Challenges**
 - Let's the team managing the applications access...
 - ... keeping the control of the cluster
- **Solutions**
 - One namespace per team
 - Delegate the Routes and TLS certificates

Challenges

- **Challenges**
 - Let's the team managing the applications access...
 - ... keeping the control of the cluster
- **Solutions**
 - One namespace per team
 - Delegate the Routes and TLS certificates
 - Delegate the Listeners

Multi-Tenancy

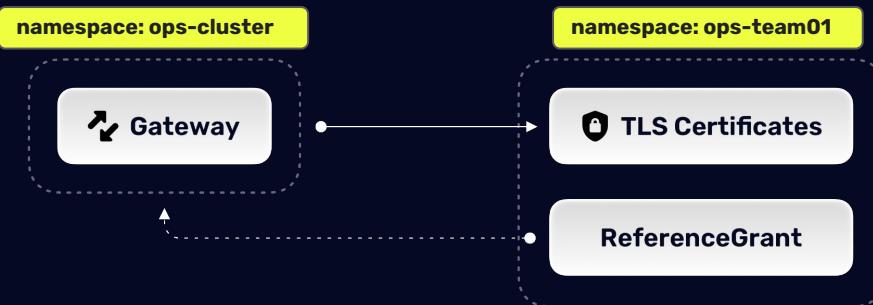
- **Challenge**
 - Sharing resources between namespaces: cross Namespace reference

Multi-Tenancy

- **Challenge**
 - Sharing resources between namespaces: cross Namespace reference
- **Solution**
 - Routes: Using namespace selectors in the listeners

Multi-Tenancy

- **Challenge**
 - Sharing resources between namespaces: cross Namespace reference
- **Solution**
 - Routes: Using namespace selectors in the listeners
 - TLS Certificates, Services: ReferenceGrant



Demo

Multi-Tenancy

- **Challenge**
 - Sharing resources between namespaces: cross Namespace reference
- **Solution**
 - Routes: Using namespace selectors in the listeners
 - TLS Certificates, Services: ReferenceGrant

How to manage lots of teams?

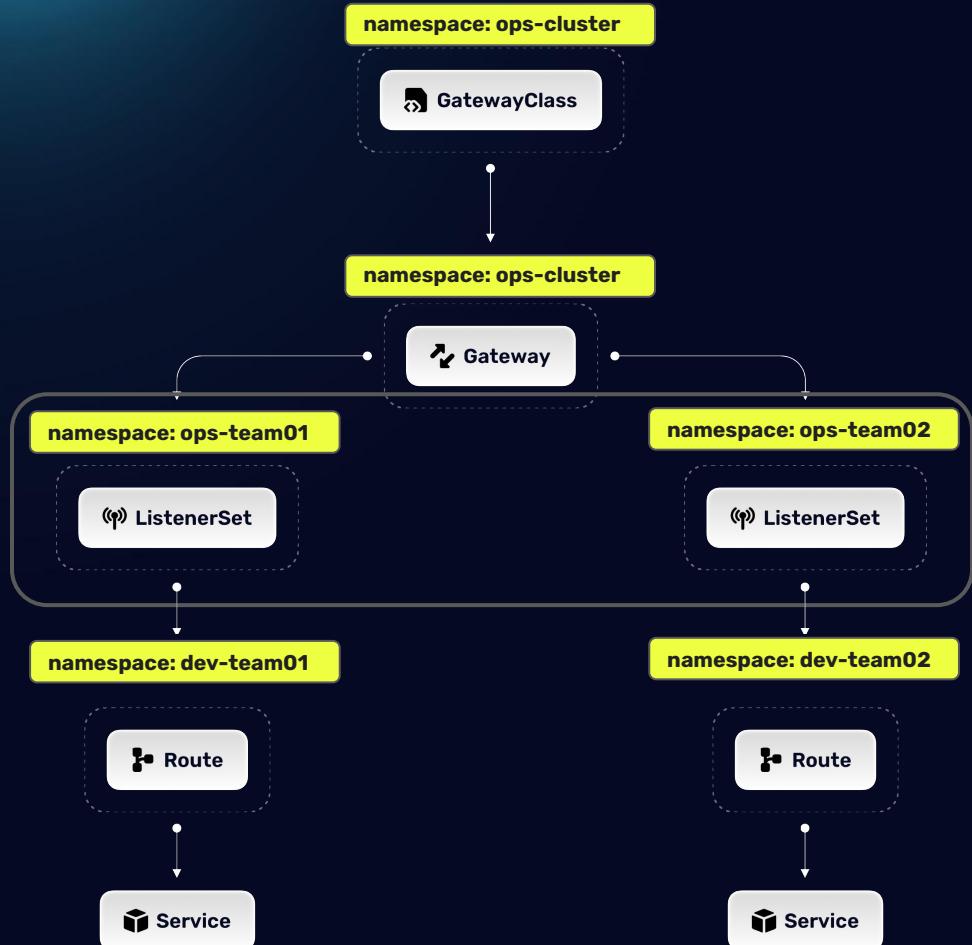
Multi-Tenancy

- **Gateway = Point of Contention**
 - Listeners
 - TLS Certificates
 - Max 64 listeners



Multi-Tenancy

- **Gateway = Point of Contention**
 - Listeners
 - TLS Certificates
 - Max 64 listeners
- **Solution**
 - ListenerSets



● ● ●

```
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: company-gateway
  namespace: ops-cluster
spec:
  gatewayClassName: traefik
  allowedListeners:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          ops-team-ns: "true"
```

● ● ●

```
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: company-gateway
  namespace: ops-cluster
spec:
  gatewayClassName: traefik
  allowedListeners:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          ops-team-ns: "true"
```

● ● ●

```
apiVersion: v1
kind: Namespace
metadata:
  name: ops-team01
  labels:
    ops-team-ns: "true"
```

```
● ● ●
apiVersion: gateway.networking.k8s.io/v1alpha1
kind: Gateway
metadata:
  name: company-gateway
  namespace: ops-cluster
spec:
  gatewayClassName: traefik
  allowedListeners:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          ops-team-ns: "true"
```

```
● ● ●
apiVersion: v1
kind: Namespace
metadata:
  name: ops-team01
  labels:
    ops-team-ns: "true"
---
apiVersion: gateway.networking.x-k8s.io/v1alpha1
kind: XListenerSet
metadata:
  name: team01-listenerset
  namespace: ops-team01
spec:
  parentRef:
    name: company-gateway
    namespace: ops-cluster
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "*.{team01}.company.io"
      allowedRoutes:
        namespaces:
          from: Selector
          selector:
            matchLabels:
              dev-team-name: "team01"
      tls:
        certificateRefs:
          - name: team01-cert
```

```
●●●
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: company-gateway
  namespace: ops-cluster
spec:
  gatewayClassName: traefik
  allowedListeners:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          ops-team-ns: "true"
```

```
●●●
apiVersion: v1
kind: Namespace
metadata:
  name: ops-team01
  labels:
    ops-team-ns: "true"
---
apiVersion: gateway.networking.k8s.io/v1
kind: XListenerSet
metadata:
  name: team01-listenerset
  namespace: ops-team01
spec:
  parentRef:
    name: company-gateway
    namespace: ops-cluster
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "*.{team01}.company"
    allowedRoutes:
      namespaces:
        from: Selector
        selector:
          matchLabels:
            dev-team-name: "team01"
  tls:
    certificateRefs:
      - name: team01-cert
```

```
●●●
apiVersion: v1
kind: Namespace
metadata:
  name: dev-team01
  labels:
    dev-team-name: "team01"
```

```
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: company-gateway
  namespace: ops-cluster
spec:
  gatewayClassName: traefik
  allowedListeners:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          ops-team-ns: "true"
```

```
apiVersion: v1
kind: Namespace
metadata:
  name: ops-team01
  labels:
    ops-team-ns: "true"
---
apiVersion: gateway.networking.k8s.io/v1
kind: XListenerSet
metadata:
  name: team01-listenerset
  namespace: ops-team01
spec:
  parentRef:
    name: company-gateway
    namespace: ops-cluster
  listeners:
    - name: websecure
      port: 8443
      protocol: HTTPS
      hostname: "*.team01.company.io"
  allowedRoutes:
    namespaces:
      from: Selector
      selector:
        matchLabels:
          dev-team-name: "team01"
  tls:
    certificateRefs:
      - name: team01-cert
```

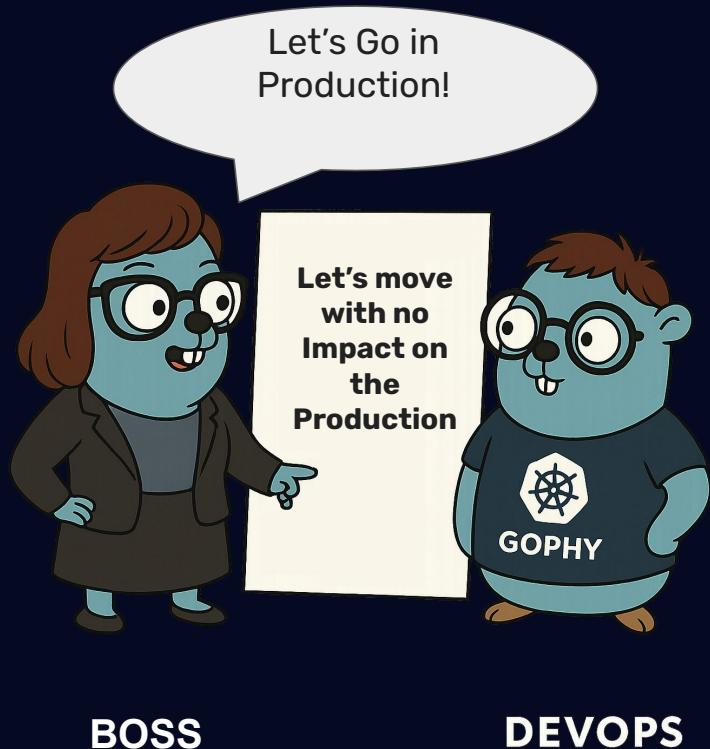
```
apiVersion: v1
kind: Namespace
metadata:
  name: dev-team01
  labels:
    dev-team-name: "team01"
---
kind: HTTPRoute
apiVersion: gateway.networking.k8s.io/v1
metadata:
  name: httproute-whoami
  namespace: dev-team01
spec:
  parentRefs:
    - name: team01-listenerset
      kind: XListenerSet
      group: gateway.networking.k8s.io
      namespace: ops-team01
  # Define the hostname
  hostnames:
    - "myapp.team01.company.io"
  rules:
    - backendRefs:
        - name: whoami
          namespace: team01
          port: 80
          kind: Service
          group: ""
```



From Ingress to Gateway API

Migration

- Challenges
 - Seamless migration
 - 0 downtime



Migration

- Challenges
 - Seamless migration
 - 0 downtime
- Solution: Traefik
 - Multi K8S providers
 - Route-by-route migration



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

Demo



There is a catch...

An Inevitable Outcome

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



**Ingress NGINX
Retirement in
March 2026**

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



Critical Notice

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

49

DAYS

07

HOURS

18

MINUTES

04

SECONDS

→ ingressnginxmigration.org





Thank you!

 emilevauge

 nmengin

 træfiklabs

