Calico Gateway API

Month Year

Presented by *Solutions Architect*

TIGERA

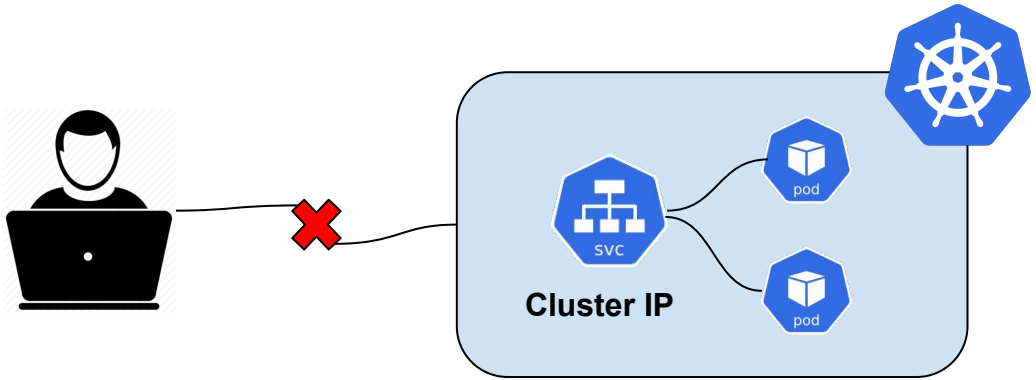
Agenda

- Kubernetes' Native Limitations
- Ingress Features
- Legacy Ingress API vs Gateway API
- Calico Ingress Gateway

TIGERA

Why Kubernetes Needs an Ingress Controller

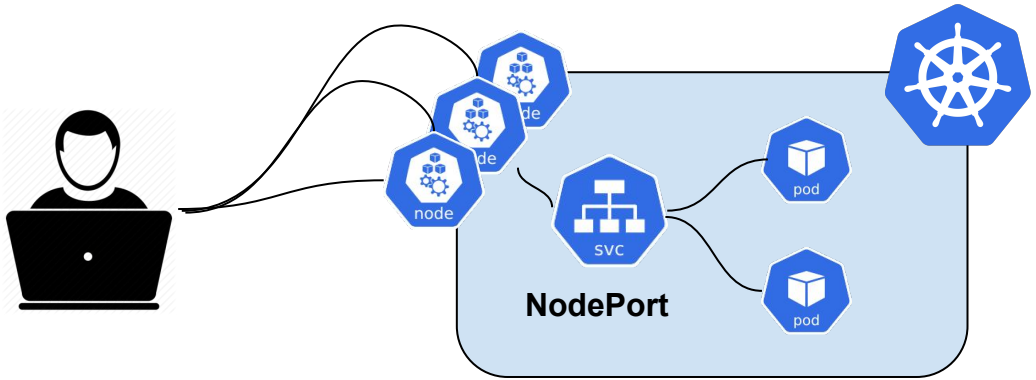
- Kubernetes provides three core service types for exposing apps:
  - ClusterIP
  - NodePort
  - LoadBalancer
- Each solves a piece of the puzzle, but gaps remain.



- Internal-only access
- No HTTP-aware routing
- Manual TLS management

Why Kubernetes Needs an Ingress Controller

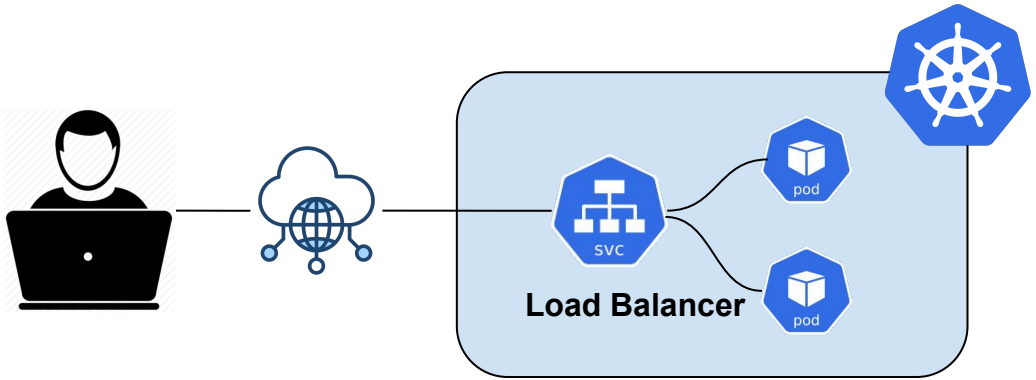
- Kubernetes provides three core service types for exposing apps:
  - ClusterIP
  - NodePort
  - LoadBalancer
- Each solves a piece of the puzzle, but gaps remain.



- Exposes apps on a static port (30000-32767) on *every node*.
- Security risks
  - Open firewall rules for high ports.
- No load balancing
  - Clients must handle node failures
- Port conflicts
  - Only one service per port across the cluster

Why Kubernetes Needs an Ingress Controller

- Kubernetes provides three core service types for exposing apps:
  - ClusterIP
  - NodePort
  - LoadBalancer
- Each solves a piece of the puzzle, but gaps remain.

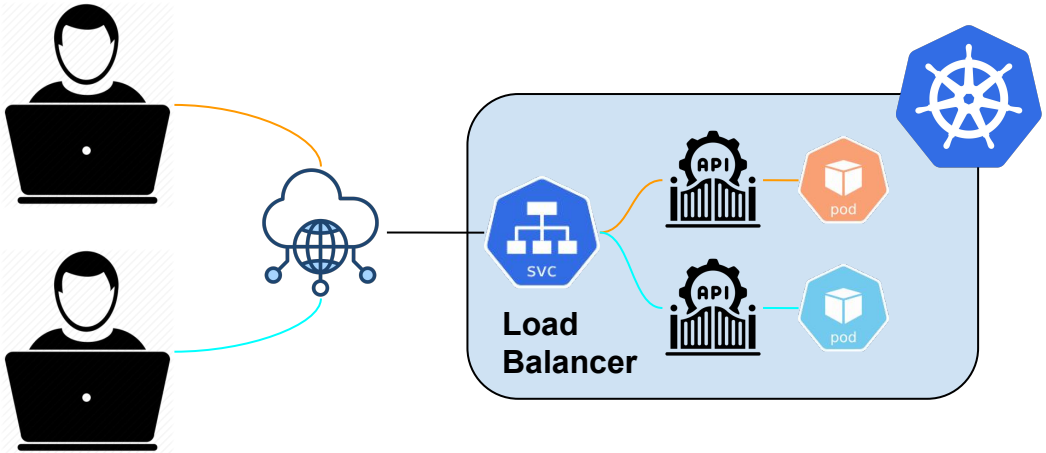


- Cloud or Vendor specific
  - Requires AWS/GCP/Azure.
  - Needs Integration On-prem.
- Costly and wasteful
  - Provisions one LB per service ($$$).
- Still no HTTP features
  - L4 (TCP) only unless using cloud-specific add-ons.

Why Kubernetes Needs an Ingress Controller

So how do we solve this?

Enter **Ingress Controllers** the Swiss Army knife for Kubernetes traffic.



- Requires a Load Balancer but can be used for as many services you need.
- Single entrypoint for all HTTP/S traffic.
- Portable (works anywhere K8s runs)
- Rich features (TLS, routing, auth, etc.)

Ingress Features: Solutions for Business

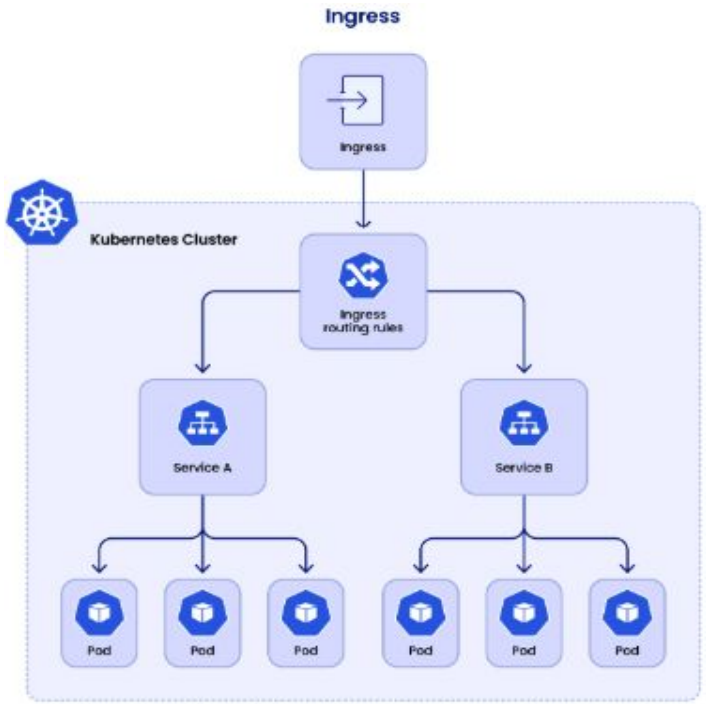
External Access Management
Kubernetes Service (ClusterIP) is internal-only; Ingress exposes HTTP/HTTPS routes externally.

Traffic Routing Rules
Path-based routing (/api → backend-service), host-based routing (app.example.com).

TLS Termination
Handles SSL/TLS certificates (e.g., via Let's Encrypt with cert-manager).

API Gateway Features
Rate limiting, authentication, request rewriting.

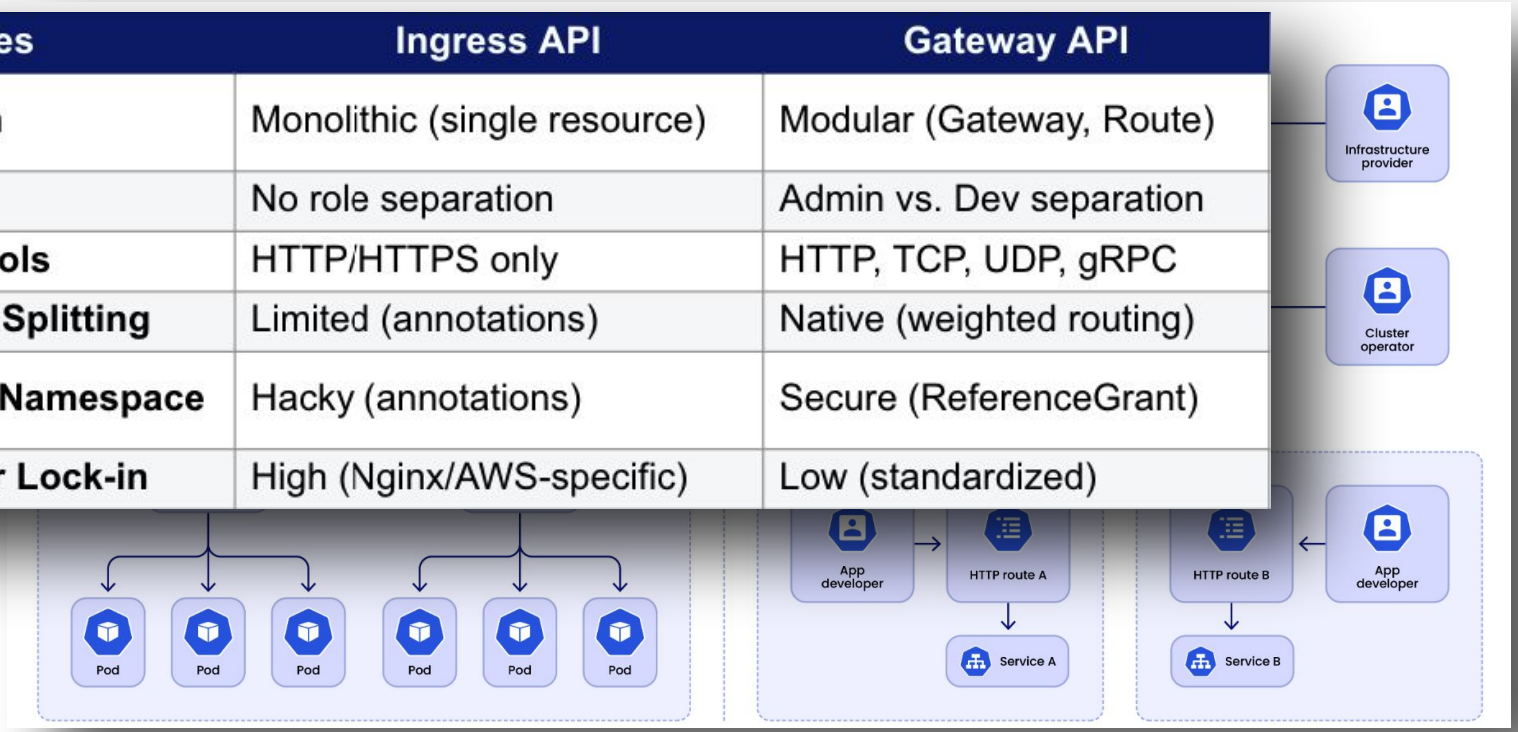
Standardization vs. Cloud LBs
Avoids cloud-specific Load Balancers (AWS ALB, GCP LB) for portability.



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

Calico Ingress Gateway

Envoy-Based Foundation

Powered by Envoy Proxy: Leverages Envoy's battle-tested L7/L4 data plane for high performance and observability.

Gateway API-Native: Implements Kubernetes Gateway API standards (e.g., HTTPRoute, TCPRoute).

Extensions for Production (CRDs)

Traffic Control: ClientTrafficPolicy, BackendTrafficPolicy (timeouts, retries).

Security: SecurityPolicy (mTLS, WAF integration).

Extensibility: EnvoyExtensionPolicy (Wasm filters, custom logic).

Production-Grade Features

Traffic Management: Rate limiting, canary releases, circuit breaking.

Security: TLS termination, OIDC authentication, DDoS protection.

Observability: Metrics (Prometheus), logs, and distributed tracing.

Calico Ingress Gateway

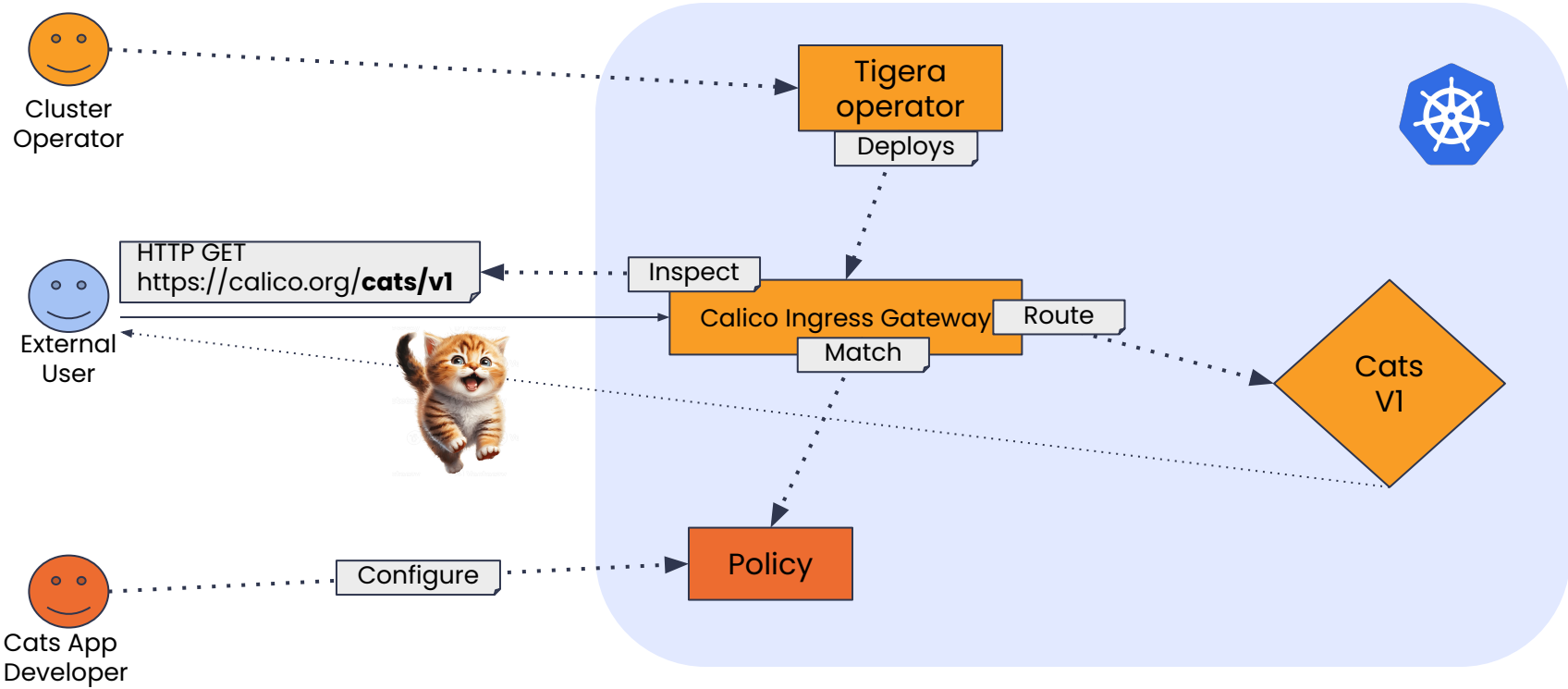
Tigera - Creator and Maintainer of Calico
Backed by Tigera's enterprise support, professional services, and deep engineering expertise for Calico's Envoy Gateway.

Leverage custom GatewayClasses for granular control
Custom GatewayClasses let you tailor gateway behavior (e.g., rate limiting, access control) to your infrastructure's specific needs.

Combine traffic mirroring with traffic splitting for safer deployments
Combine traffic mirroring + splitting in HTTPRoute to safely test new versions before full rollout.

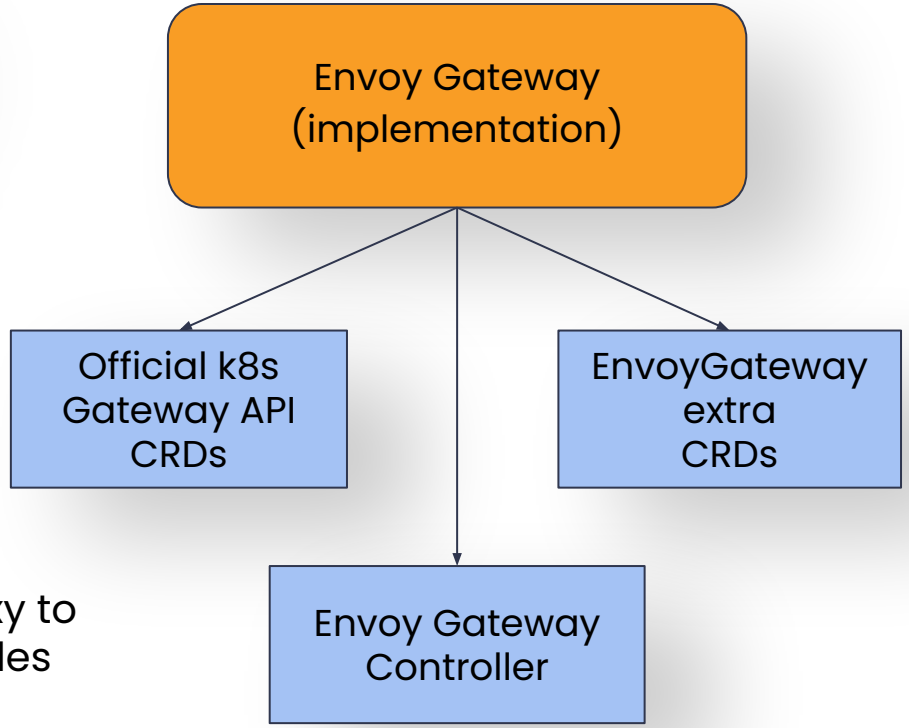
Use Gateway API for multi-cluster setups
Deploy Gateway API across clusters for centralized, scalable multi-cluster traffic management.

What is Calico Ingress Gateway?



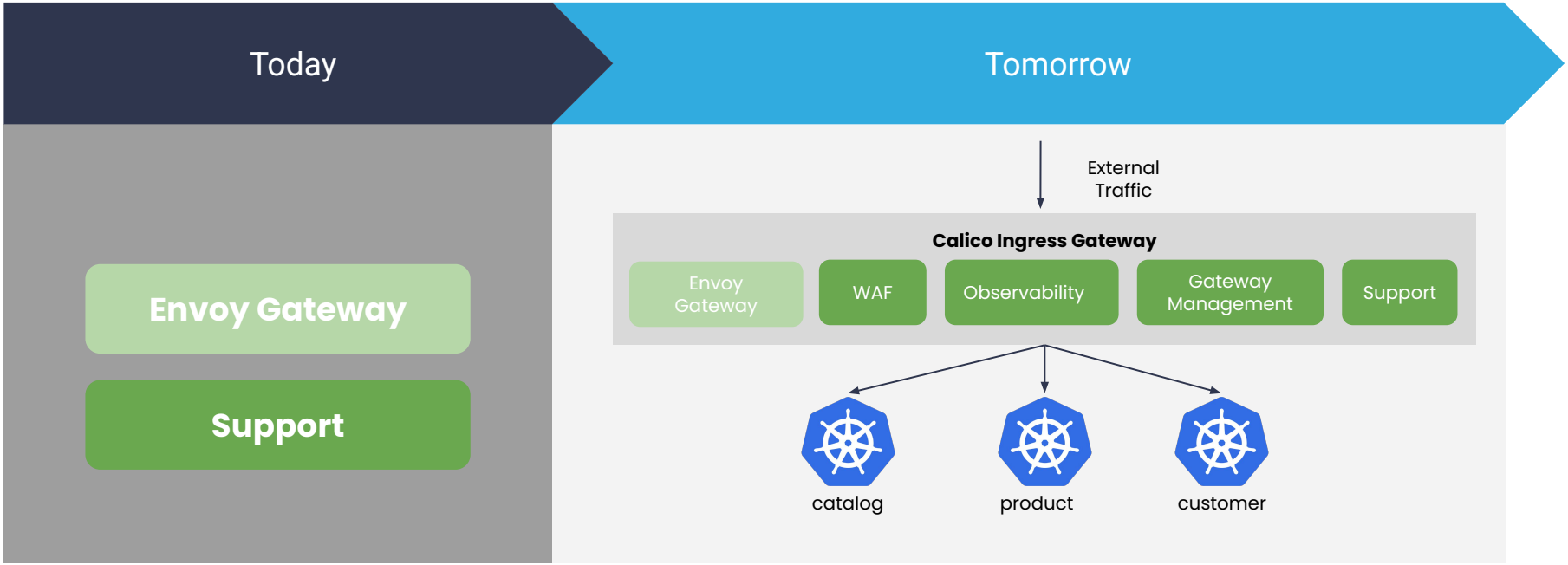
Our Implementation

```
kind: GatewayAPI
apiVersion: operator.tigera.io
metadata:
  name: tigera-secure
spec:
  ... customizations ...
```



Calico Ingress Gateway requires a proxy to route to external traffic according to rules hence the use of Envoy Proxy

Calico Ingress Gateway Roadmap



Thank you

TIGERA

Follow us on: