

Securing K8s APIs with Emissary-Ingress

October 5th, 2022

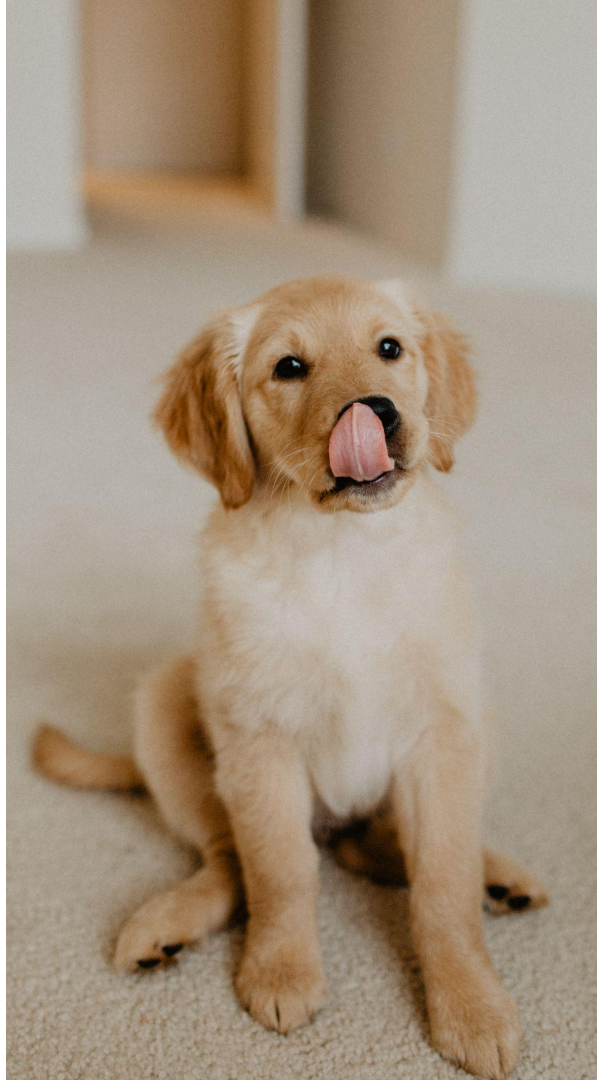
Today's Agenda

K8s Ingresses What is the K8s Ingress API and what are the limitations?

Alternatives Can we get better?

Demo time Let's get dirty!

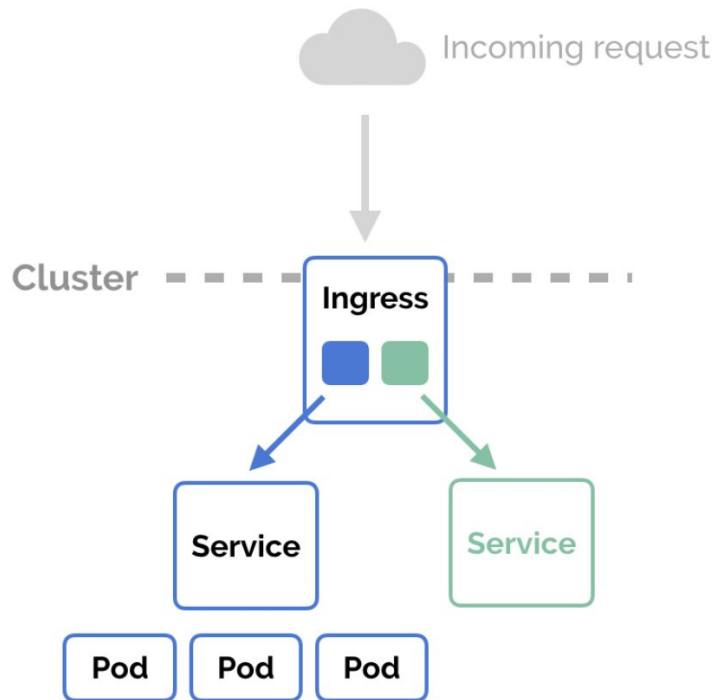
Q&A I ask, you answer :)



K8s Ingress

What is Ingress?

Ingress



Ingress API capabilities

- Gives services externally-reachable URLs
- Load balances traffic
- Terminates SSL/TLS
- Name-based virtual hosting
- Provides decentralised configuration for everything mentioned above
- API reference → <https://kubernetes.io/docs/reference/kubernetes-api/service-resources/ingress-v1/>



Ingress API missing out

- Limited traffic management capabilities (some rewrites and redirects via annotations)
- No retries, circuit breakers etc.
- Doesn't allow rate limiting specifications
- Cannot enforce authentication or authorization for Ingress routes
- No specs for canary or traffic shadowing



Alternatives

Can we get better?

API Gateways

Why?

- The Ingress API is quite basic when it comes to security and advanced traffic management capabilities

How?

- Kong → if you love Nginx & Lua scripting - <https://konghq.com/>
- Apache APISIX → a newcomer to the open source community - <https://apisix.apache.org/>
- Tyk → K8s native written in Go - <https://github.com/TykTechnologies/tyk>
- Ocelot → if you need to stick with the .Net stack - <https://github.com/ThreeMammals/Ocelot>
- Express Gateway → built around Express.js - <https://www.express-gateway.io/>
- Solo Api Gateway (formerly Gloo) → based on the Envoy reverse proxy - <https://github.com/solo-io/gloo>
- Emissary Ingress (formerly Ambassador) → based on the Envoy reverse proxy - <https://github.com/emissary-ingress/emissary>
- Traefik → K8s native written in Go - <https://github.com/traefik/traefik>
- Spring Cloud Gateway → if you're on the JVM stack and want a seamless transition - <https://spring.io/projects/spring-cloud-gateway>
- And many more for sure :)

What?

- So by using an API Gateway that has Ingress Controller capabilities, we can leverage the benefits of both worlds



You are the chosen one!

The list is long but I choose you Emissary-Ingress:

- It's based on the well-known reverse proxy Envoy → same technology on which Istio is built
- Adheres to the Ingress API so can work as an Ingress Controller
- Can gather and export route based metrics and dump into Prometheus or StatsD style metrics
- Integrates with OpenTracing API → Jaeger and Zipkin compatible
- Support for Rate Limiting & Authentication as external services
- Provides an out of the box production ready configuration for Envoy → while still being flexible enough to allow fine-grained configuration
- Uses CRDs and a decentralised configuration model → each team can manage interaction between the API Gateway and their service without much knowhow of how it's configured
- Works really well along service meshes such as Istio, Consul or Linkerd



Demo time

GitHub repo

<https://bit.ly/3CuzGr6>



Thank You!

What Questions Can I Answer?

You can stalk me on

Linkedin - <https://www.linkedin.com/in/bogdanmariesan/>

Github - <https://github.com/bmariesan/>

Twitter - @bogdan32x

