Migrating from the Ingress API to the Gateway API

Nigel Brown

Freelance Technical Author

@n_brownuk | @nigelb@fosstodon.org | windsock.io

Conversion

Existing Ingress API object definitions require translating to Gateway API object definitions.

Traffic Entry Points

Ingress Controller

Provides an HTTP entry point to the cluster, often shared by all ingress objects

Gateway Listener

Listeners explicitly define entry points for ingress traffic that match a profile

Data Plane Processing

Ingress API

Ingress definitions specify an IngressClass which is associated with a controller

Gateway API

HTTPRoute definitions use the ParentRefs field to refer to a parent Gateway

HTTP Routing

Virtual host routing

- An Ingress API rule with a ‘host’ field enables routing based on HTTP header
- HTTPRoute ‘hostnames’ provide a similar capability

Path-based routing

- HTTP paths in Ingress API rules are directly analogous to HTTPRoute rules

Default Backend

Ingress API allows for defining a ‘catch all’ route to a backend service

A ‘catch all’ route and service needs to be explicitly defined with the Gateway API

Ingress to Gateway

Ingress2gateway helps translate Ingress API resources to Gateway API resources, specifically HTTPRoutes.

Kubernetes SIG-Network subproject, https://github.com/kubernetes-sigs/ingress2gateway

Demo

Converting an Ingress Definition to the Gateway API

- An ingress object has already been created
- Use the ‘ingress2gateway’ utility to create equivalent Gateway API definitions

Course Summary

What we covered:

- Historical ingress APIs have limitations
- Gateway API provides a richer ingress experience with extensibility
- Allows the separation of concerns
- Migration from Ingress API is necessary

Up Next:

Role-based Configuration of Ingress in Kubernetes Clusters

thanks!
Tombow
Acid Free
ABT
N15