KCD
GROWING CLOUD NATIVE TOGETHER
INDONESIA 2024

Google Cloud | NUTANIX | Red Hat | Alibaba Cloud | aws

Introduction to Kubernetes gateway api

Zufar Dhiyaulhaq

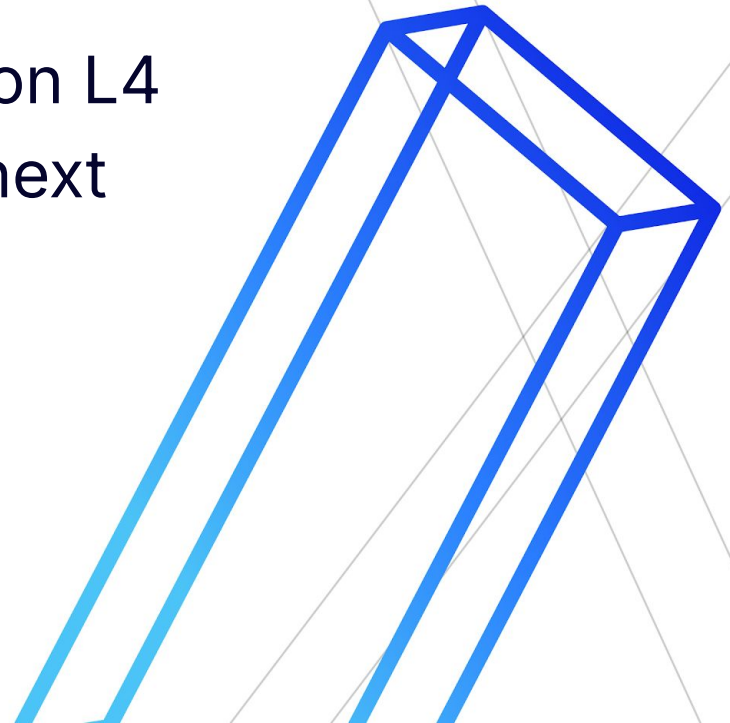
Engineering Manager
@ GoTo Financial

Ananda Dwi Rahmawati

Cloud & DevOps
Engineer @ Singapore

What is the Kubernetes Gateway API?

A Kubernetes project focused on L4 and L7 routing in Kubernetes, next generation of **Ingress**



What is the Kubernetes Gateway API?

Standardized Traffic Routing

Provides a consistent and vendor-neutral way to configure and manage traffic routing within Kubernetes clusters.

Advanced Traffic Management

Offers a rich set of features for advanced traffic management like traffic splitting or mirroring, Regex matching, header based matching, TLS, gRPC, etc.

Extensible and Customizable

It designed to be extensible, from users deployments model, and even custom functionality on the implementation of Gateway API

Use Cases

Basic north/south

Multiple applications behind a single Gateway

Gateway and mesh*

- East/West traffic*

* Experimental

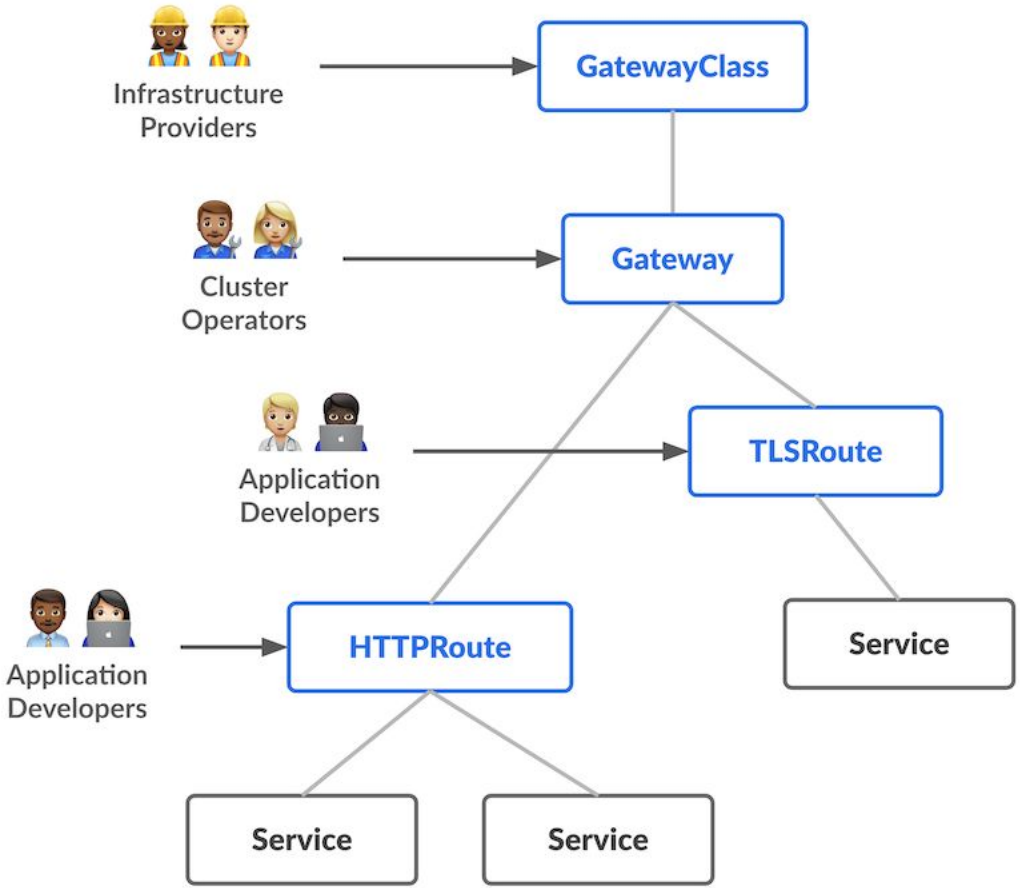
Why Gateway API?

Ingress resources is **to simples** to manage advanced use cases

Advanced use cases is implemented via annotations, non-standard approach leads to **fragmentation** across Ingress Controllers

Portability challenging due to non-standard approach

CRDs


Infrastructure Providers
GatewayClass
Cluster Operators
Gateway
Application Developers
TLSRoute
Application Developers
HTTPRoute
Service
Service
Service

GatewayClass

- Represent a class of Gateways that can be instantiated.
- **Template that can be used to construct** the real gateway pods

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: internal-gateway
spec:
  controllerName: gateway.envoyproxy.io/gatewayclass-controller
  parametersRef:
    group: gateway.envoyproxy.io
    kind: EnvoyProxy
    name: internal-proxy-config
    namespace: envoy-gateway-system
```

```yaml
apiVersion: gateway.envoyproxy.io/v1alpha1
kind: EnvoyProxy
metadata:
  name: internal-proxy-config
  namespace: envoy-gateway-system
spec:
  provider:
    type: Kubernetes
    kubernetes:
      envoyService:
        type: LoadBalancer
        annotations:
          service.beta.kubernetes.io/alibaba-cloud-loadbalancer-address-type: "intranet"
```

Gateway

- Creating the underlying gateway infrastructure based on the template on GatewayClass.

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: internal-gateway
spec:
  controllerName: gateway.envoyproxy.io/gatewayclass-controller
  parametersRef:
    group: gateway.envoyproxy.io
    kind: EnvoyProxy
    name: internal-proxy-config
    namespace: envoy-gateway-system
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: api-internal-kubernetes-com-gw
  namespace: envoy-gateway-system
spec:
  gatewayClassName: internal-gateway
  listeners:
    - name: http
      port: 8080
      protocol: HTTP
      hostname: api.internal.kubernetes.com
```

Gateway

Depending on the **GatewayClass**, the creation of a **Gateway** could do any of the following actions:

- Use cloud APIs to create an LB instance.
- Spawn a new instance of a software LB (in this or another cluster).
- Add a configuration stanza to an already instantiated LB to handle the new routes.
- Program the SDN to implement the configuration.
- Something else we haven't thought of yet...

HTTPRoute

- Handle various types of network traffic with detailed matching rules.

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: backend-route
  namespace: envoy-gateway-system
spec:
  parentRefs:
    - name: api-internal-kubernetes-com-gw
      namespace: envoy-gateway-system
  rules:
    - backendRefs:
        - kind: Service
          name: backend
          namespace: gateway-api-service
          port: 3000
          weight: 100
      matches:
        - path:
            type: PathPrefix
            value: /
```

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: api-internal-kubernetes-com-gw
  namespace: envoy-gateway-system
spec:
  gatewayClassName: internal-gateway
  listeners:
    - name: http
      port: 8080
      protocol: HTTP
      hostname: api.internal.kubernetes.com
```

Migrating Ingress to Gateway API

Ingress to Gateway

Ingress2gateway helps translate Ingress and provider-specific resources (CRDs) to Gateway API resources. Ingress2gateway is managed by the Gateway API SIG-Network subproject.

https://github.com/kubernetes-sigs/ingress2gateway

THANK YOU