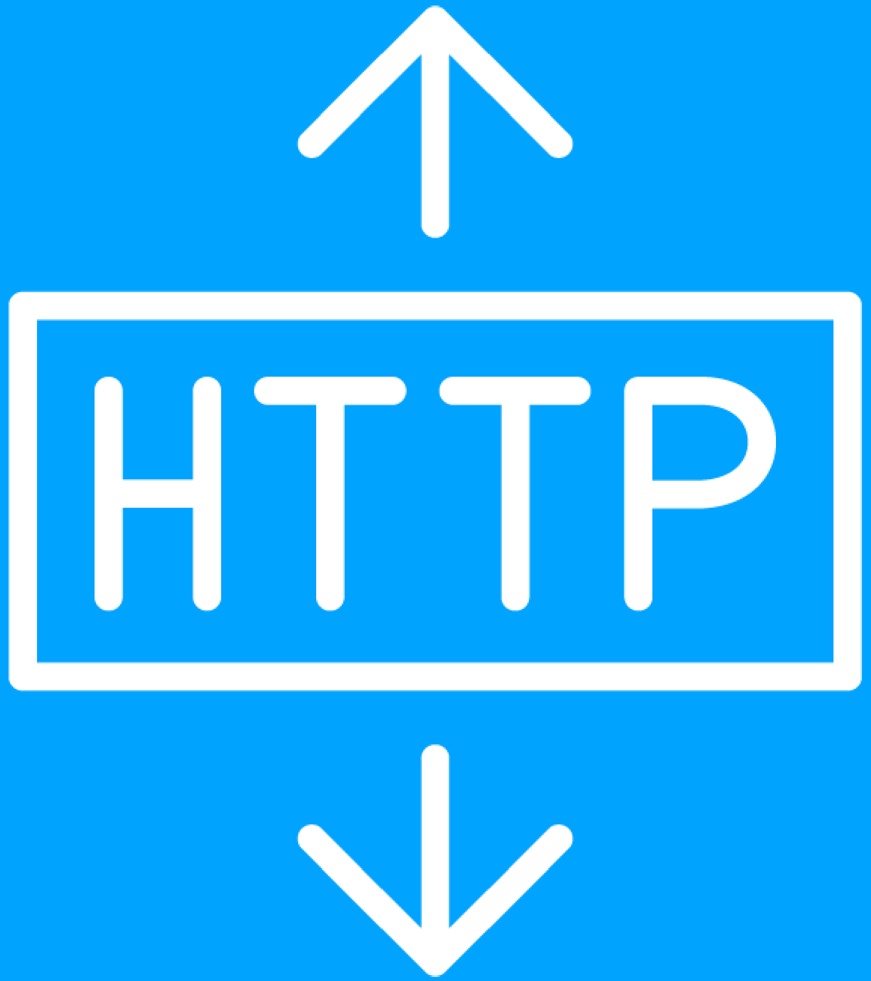
Routing HTTP Traffic to Backend Services

Nigel Brown

Freelance Technical Author

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



HTTP Traffic Routing

Gateway API supports multiple protocols, but our focus is on routing HTTP traffic.

GatewayClass

A resource that describes the common behavior of a set of gateways used for ingress purposes in Kubernetes.

Characteristics of the GatewayClass

Provides a representation of a gateway controller running in the cluster

Object instances of GatewayClass resources are scoped to the cluster

Decouples those who define application traffic routes from the implementation

Gateway Controller Types

Public Cloud

Provisions cloud resources that enable HTTP routing to backend services

Proxies

Configures proxy software to facilitate HTTP routing to backend services

Defining a GatewayClass

The ‘controllerName’ is a required field.

gatewayclass.yaml

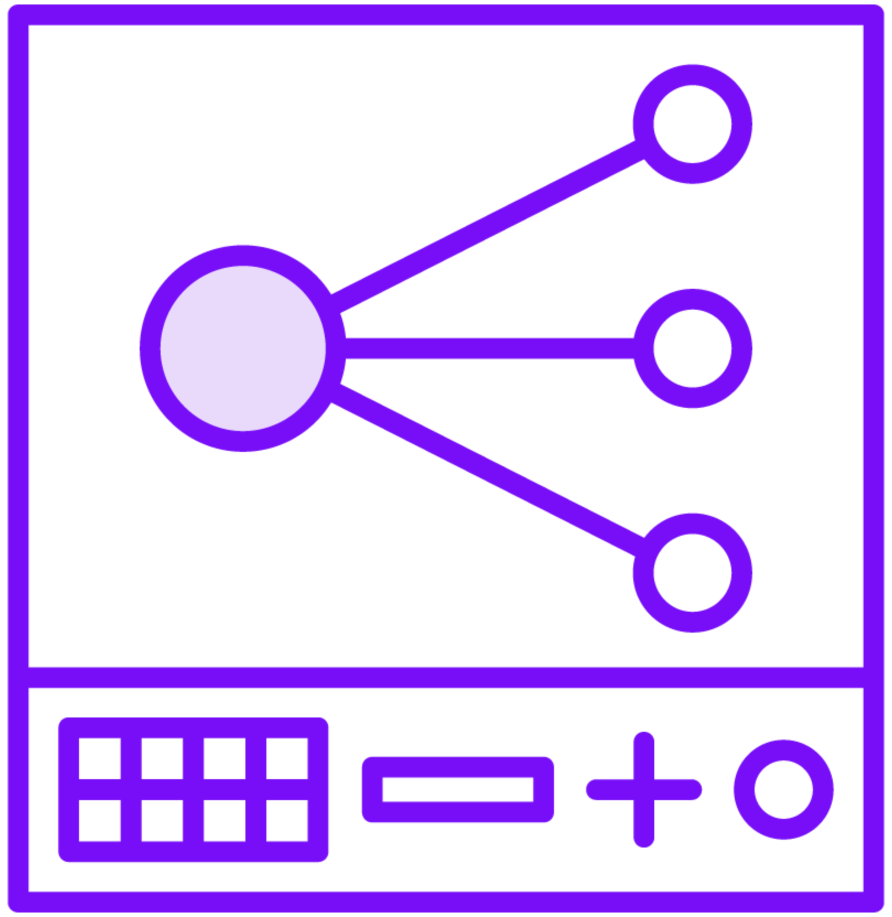
```
---
apiVersion: gateway.networking.k8s.io/v1
kind: GatewayClass
metadata:
  name: envoy-gateway-dev
spec:
  controllerName: gateway.envoyproxy.io/gw-controller
```

```
$ kubectl get gatewayclasses
NAME                CONTROLLER                            ACCEPTED    AGE
envoy-gateway-dev   gateway.envoyproxy.io/gw-controller   True        9s
```

Gateway

A Gateway describes how traffic can be translated to backend services within the cluster.

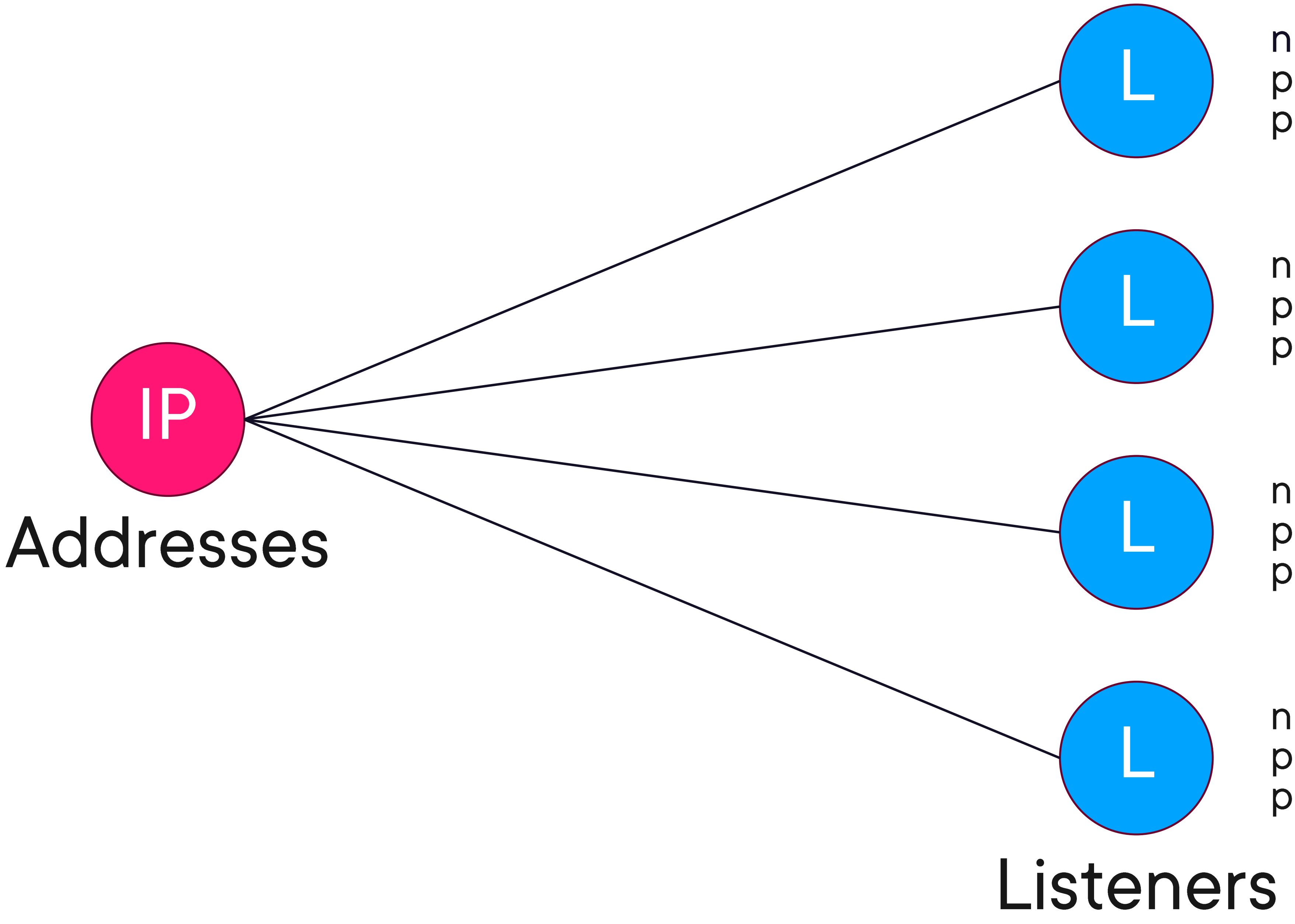
Consequences of Gateway Object Creation



What happens when Gateway objects are created?

- Gateway ‘managed’ by GatewayClass
- Triggers configuration of infrastructure
- Actions performed by gateway controller

Binding Listeners to Addresses



name: **web-app**
port: **80**
protocol: **HTTP**

name: **tcp-app**
port: **8080**
protocol: **TCP**

name: **secure-web-app**
port: **443**
protocol: **HTTPS**

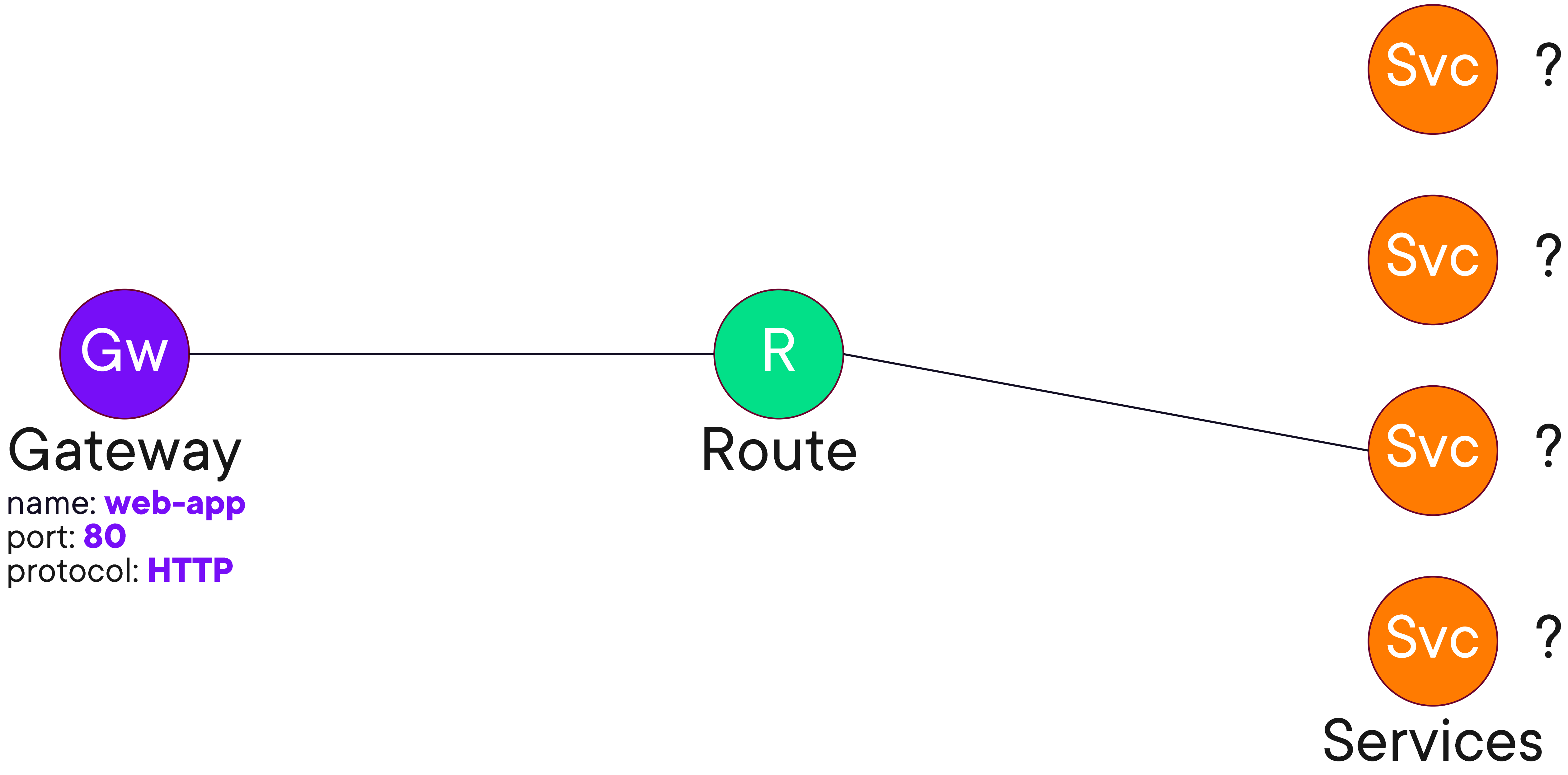
name: **grpc-app**
port: **5000**
protocol: **HTTPS**

Gateway Object

gateway.yaml

```yaml
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: web-app-dev-gw
spec:
  gatewayClassName: envoy-gateway-dev
  listeners:
    - protocol: HTTP
      port: 80
      name: http
      hostname: dev.mycorp.com
```

Routes



name: **web-app**
port: **80**
protocol: **HTTP**

HTTPRoute

An HTTPRoute is a Gateway API type for specifying routing behavior of HTTP requests from a Gateway listener to an API object, i.e. Service.

```yaml
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: web-app
spec:

<snip>
```

Defining an HTTPRoute

Referencing Gateways

An HTTPRoute can reference one or more Gateway parents.

httproute.yaml

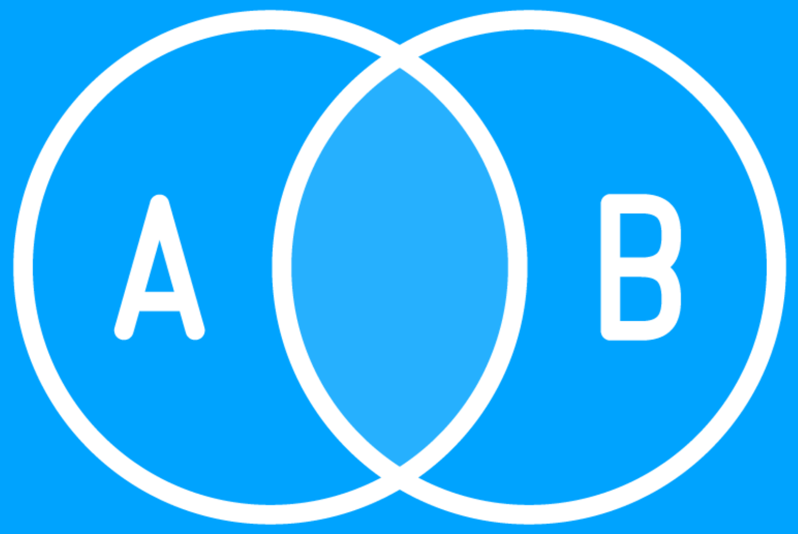
```yaml
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: web-app
spec:
  parentRefs:
    - name: web-app-dev-gw
      group: gateway.networking.k8s.io
      kind: Gateway
```

Virtual Hosts

HTTPRoutes can specify hostnames which must match HTTP host header.

httproute.yaml

```yaml
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: web-app
spec:
  parentRefs:
    - name: web-app-dev-gw
  hostnames:
    - dev.mycorp.com
```



Hostname Intersection

A request can be routed if there is an intersection between the hostnames in the Gateway Listener and HTTPRoute.

Hostnames

Example #1

Listener	HTTPRoute	
dev.mycorp.com	-	✓
	dev.mycorp.com	✓
	*.mycorp.com	✓
	qa.mycorp.com	✗
	mycorp.com	✗

Example #2

Listener	HTTPRoute	
*.mycorp.com	-	✓
	dev.mycorp.com	✓
	*.mycorp.com	✓
	qa.mycorp.com	✓
	mycorp.com	✗

Rules

httproute.yaml

```yaml
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: web-app
spec:
  parentRefs:
    - name: web-app-dev-gw
  hostnames:
    - dev.mycorp.com
  rules:
    - backendRefs:
        - name: web-app-svc
          port: 80
```

Trust

Trust is established with HTTPRoute and Gateway objects co-habiting the same namespace.

Demo

Consuming a Kubernetes Application Using the Gateway API

- Use cluster with Envoy Gateway installed
- A simple application already deployed
- Inspect and apply Gateway API object definitions
- Test access to the app in a web browser

Up Next:

Migrating from the Ingress API to the Gateway API