GatewayAPI: Ingress的終局之戰

Directed by: Shawn Ho

Google Cloud

Agenda

- Mighty Ingress
- Gateway API的組成與展示
- MultiCluster Gateway API的擴展
- Gateway API的Roadmap

Mighty Ingress:
容器世界的第一個L7 LB

OSS Kubernetes Ingress

The Good

- Very widely adopted
- Has enabled the entire K8s ecosystem to shift networking left to the developers

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: cafe-ingress
spec:
  tls:
  - hosts:
    - cafe.example.com
    secretName: cafe-secret
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /tea
        backend:
          serviceName: tea-svc
          servicePort: 80
      - path: /coffee
        backend:
          serviceName: coffee-svc
          servicePort: 80
```

OSS Kubernetes Ingress

The Bad

- Lowest common denominator API
- Not very extensible (other than through annotations)
- Not very portable (anymore)
- No formal multi-tenancy

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: cafe-ingress-with-annotations
  annotations:
    nginx.org/proxy-connect-timeout: "30s"
    nginx.org/proxy-read-timeout: "20s"
    nginx.org/client-max-body-size: "4m"
    nginx.org/server-snippets: |
      location / {
        return 302 /coffee;
      }
spec:
  rules:
  - host: cafe.example.com
    http:
      paths:
      - path: /tea
        pathType: Prefix
        backend:
          service:
            name: tea-svc
            port:
              number: 80
      - path: /coffee
        pathType: Prefix
        backend:
          service:
            name: coffee-svc
            port:
              number: 80
```

Gateway API的介紹與展示: 雷神再臨

Google Cloud

Gateway API的捧油們

Kubernetes SIG-Network Specification

Contributors

Google Cloud

Kong

traefiklabs

vmware

Red Hat

solo.io

Gateway Controllers

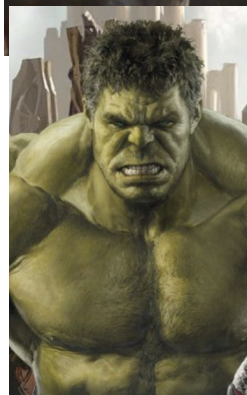
CONTOUR

traefik proxy

GKE Gateway Controller

Istio

Gloo Edge



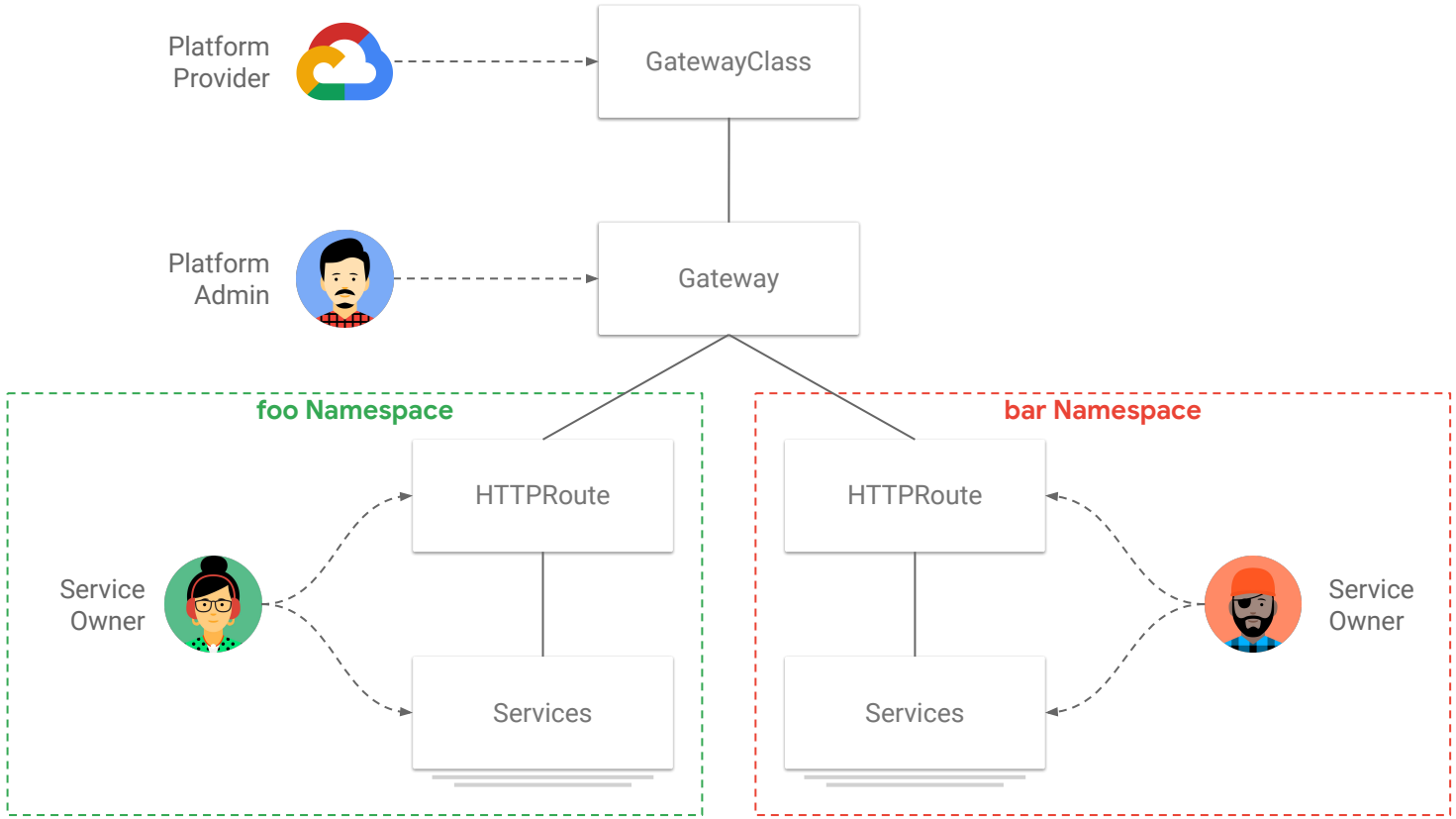
What is the Gateway API?

Role-oriented

Expressive

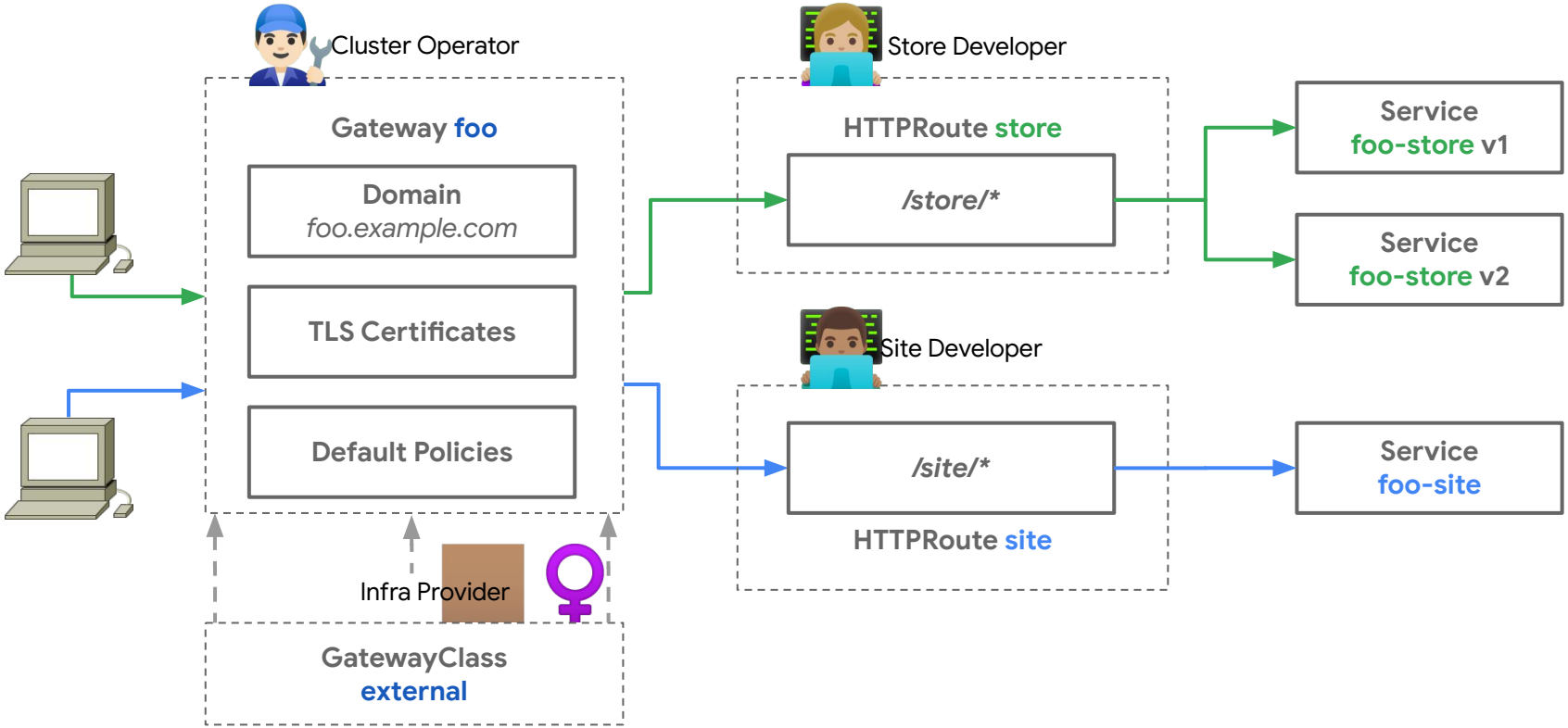
Extensible

Generic



What is the Gateway API?

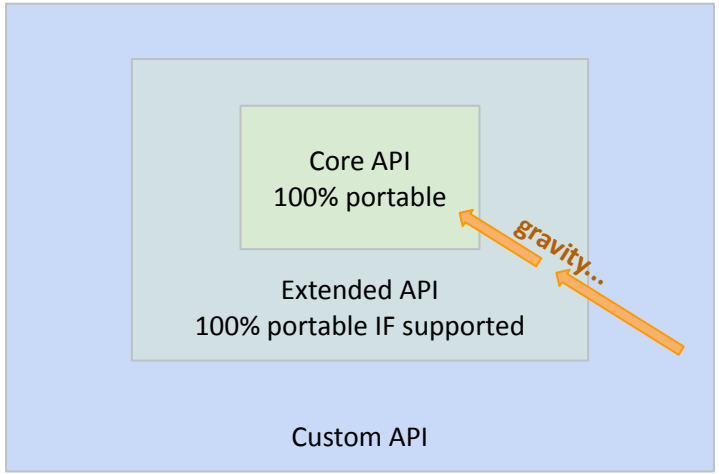
Role Oriented Resource Model



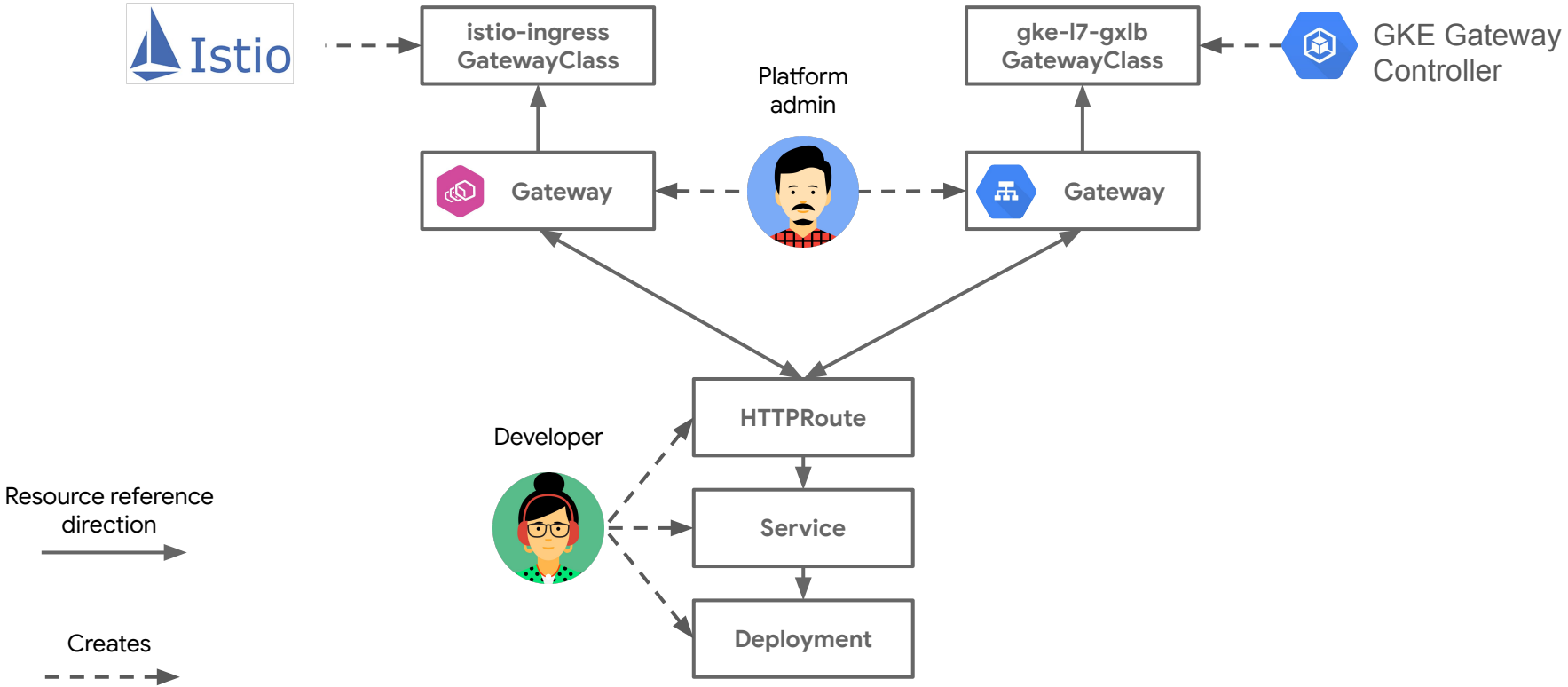
Gateway APIs have 3 Categories

- **Core**
  - MUST be supported.
  - More load balancing functionality becoming part of the core Gateway API than was in Ingress
    - Header-based matching
    - Traffic weighting
    - Regex matching
    - Health checking
    - … and more
- **Extended**
  - Feature by feature.
  - MAYBE supported, but **MUST be portable.**
  - Part of API schema.
- **Custom**
  - No guarantee for portability
  - No k8s API schema.

```
hosts:
  - hostname: team1.example.com
    rules:
      - match:
          pathType: Prefix
          path: /
          headerType: Exact
          header:
            env: stage
```



Gateway <-> Route Relationships



kind	Gateway	
apiVersion	networking.x-k8s.io/v1alpha1	
metadata.name	shared-gateway	
metadata.namespace	foobar	
spec.gatewayClassName	gke-l7-gxlb	
spec.listeners[0]	protocol	HTTPS
	port	443
	routes.kind	HTTPRoute
	tls.certificateRef.name	foo-example-com
spec.hostnames[0]	foo.example.com	
spec.rules[0]	matches[0].path.value	/*
	forwardTo.serviceName	**home**
	forwardTo.serviceNamespace	site
spec.rules1]	matches[0].path.value	/login
	forwardTo[0].serviceName	**login-v1**
	forwardTo[0].serviceNamespace	site
	forwardTo[0].weight	90
	forwardTo[1].serviceName	**login-v2**
	forwardTo[1].serviceNamespace	site
	forwardTo[1].weight	10
spec.rules[1]	matches[0].path.value	/store
	forwardTo.serviceName	**store**
	forwardTo[0].serviceNamespace	store

Platform Admin

Service Owner

Service Owner

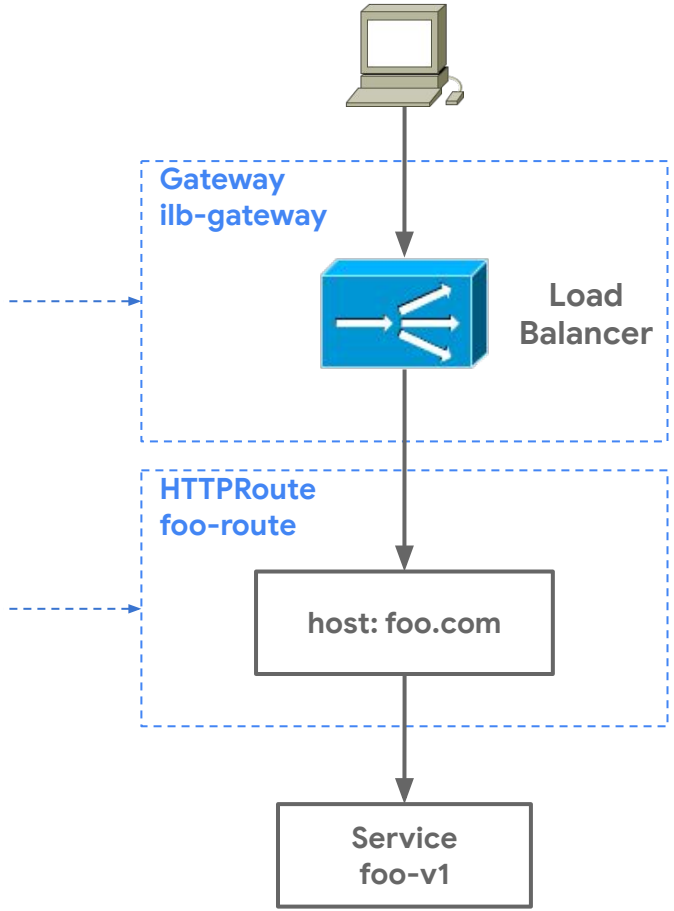
Gateway Resource

Platform admin

```
kind: Gateway
metadata:
  name: ilb-gateway
spec:
  gatewayClassName: gke-l7-rilb-mc
  listeners:
  - protocol: HTTP
    port: 80
    routes:
      kind: HTTPRoute
      selector:
        matchLabels:
          gateway: internal-gw
status:
  addresses:
  - value:  192.168.1.227
```

Developer

```
kind: HTTPRoute
metadata:
  name: foo-route
  namespace: foo
  labels:
    gateway: internal-gw
spec:
  hostnames:
  - "foo.com"
  rules:
  - forwardTo:
    - serviceName: foo-v1
      port: 8080
```



Route Binding

Platform admin

```
kind: Gateway
metadata:
  name: ilb-gateway
  namespace: infra
spec:
  gatewayClassName: gke-l7-rilb-mc
  listeners:
  - protocol: HTTP
    port: 80
    routes:
      kind: HTTPRoute
      selector:
        matchLabels:
          gateway: internal-gw
      namespaces:
        from: "All"
```

infra Namespace

foo Developer

```
kind: HTTPRoute
metadata:
  name: route3
  namespace: foo
  labels:
    gateway: internal-gw
spec:
  hostnames:
  - "foo.com"
  rules:
  - forwardTo:
    - serviceName: foo-v1
      port: 8080
```

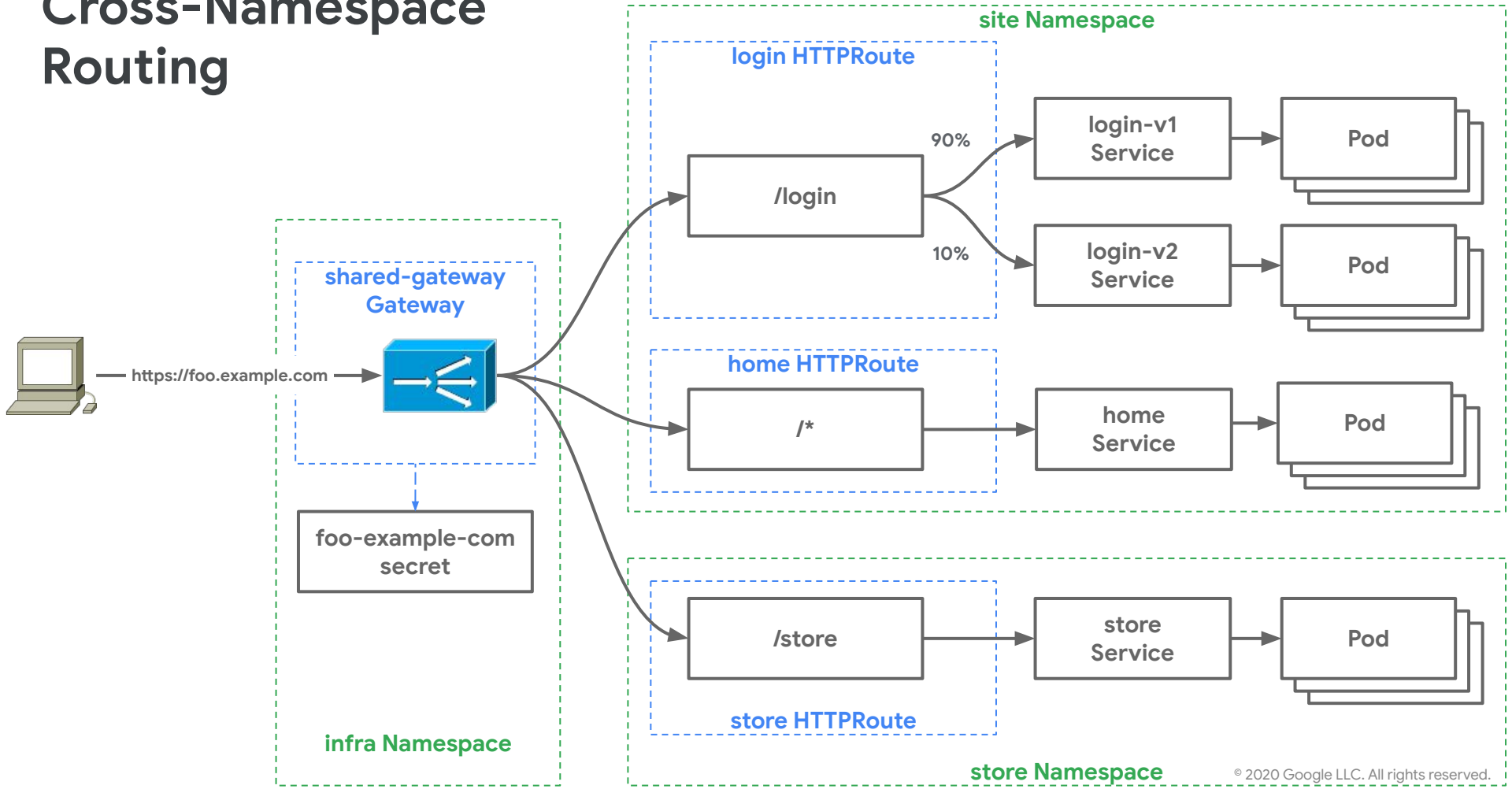
foo Namespace

```
kind: HTTPRoute
metadata:
  name: route3
  namespace: bar
  labels:
    gateway: internal-gw
spec:
  hostnames:
  - "bar.com"
  rules:
  - forwardTo:
    - serviceName: bar-v1
      port: 8080
```

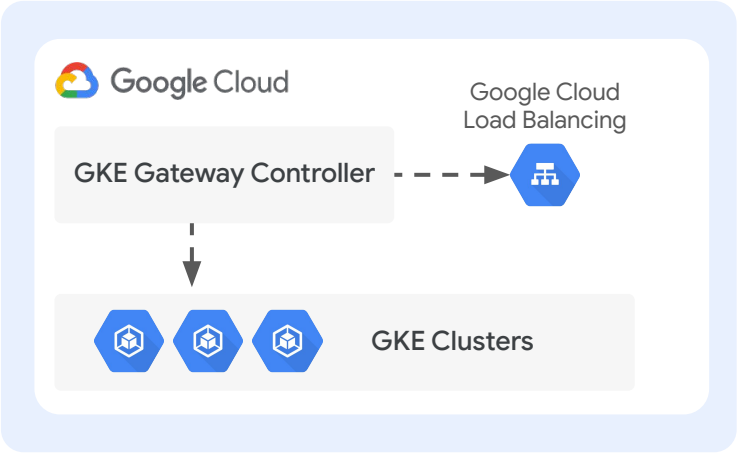
bar Developer

bar Namespace

Cross-Namespace Routing



GKE Gateway Controller

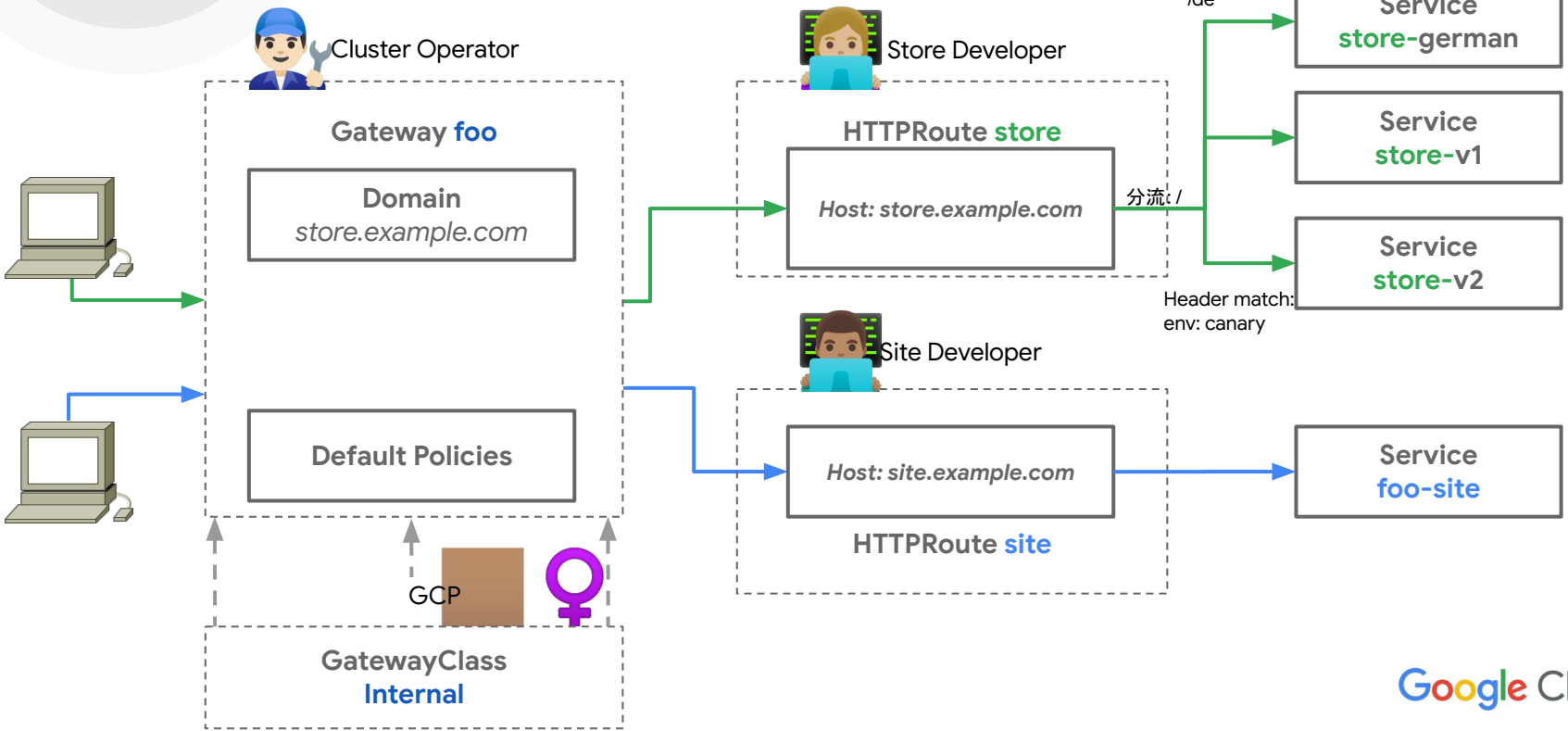


A set of Google-hosted Kubernetes controllers that orchestrate Google Cloud Load Balancers via the open source Gateway API specification.

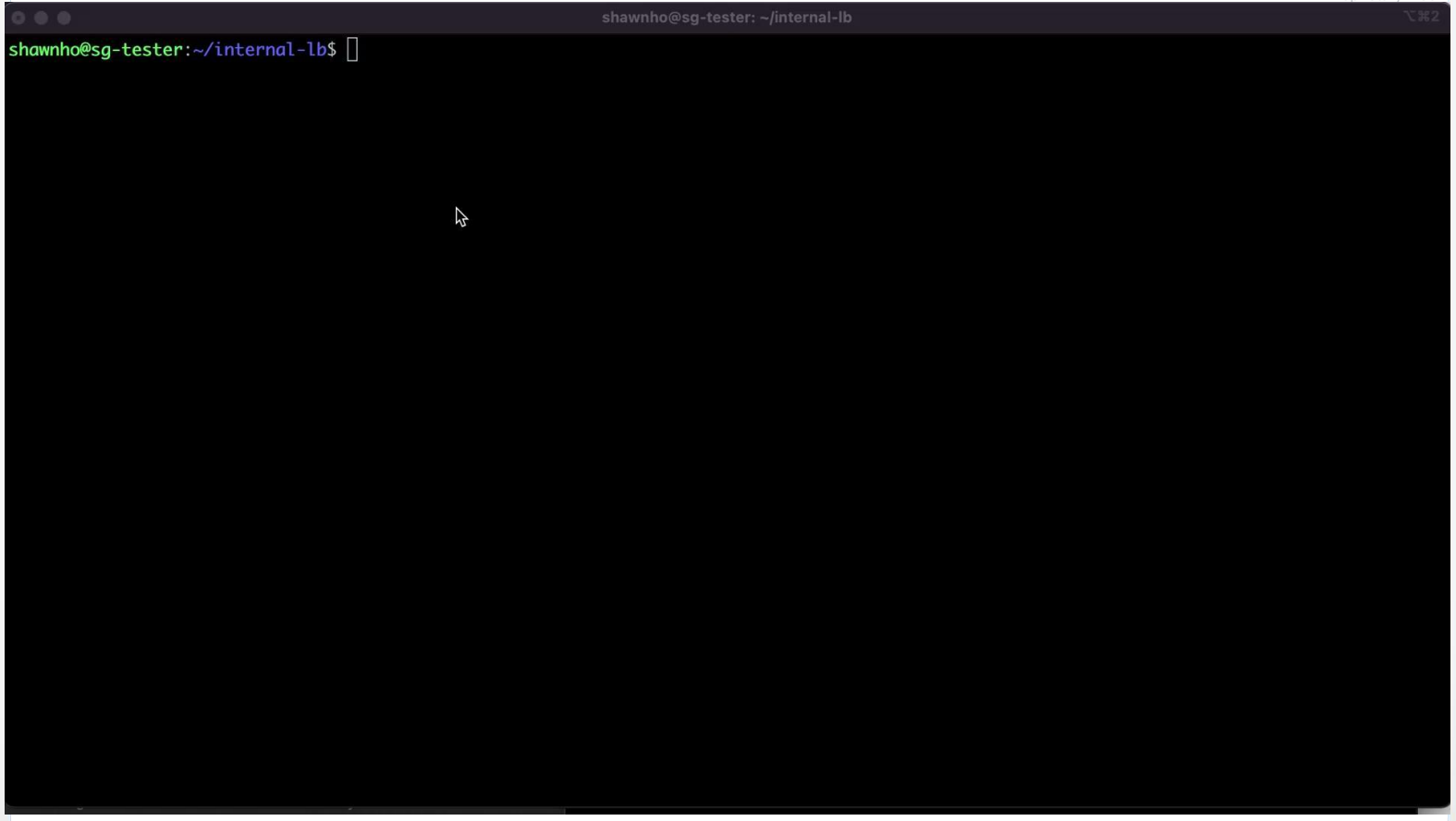
- Supports Internal and External HTTP(S) Load Balancing
- Host, path, header-based routing
- HTTP Header manipulation
- Weight-based traffic splitting
- Traffic capacity-based load balancing
- Traffic mirroring

- Multi-cluster Gateways (MCGs) for internal *and* external load balancing
- Support for CloudArmor, Identity-Aware Proxy (IAP), and Cloud CDN
- Geographic-based load balancing
- HTTP, HTTPS, HTTP/2

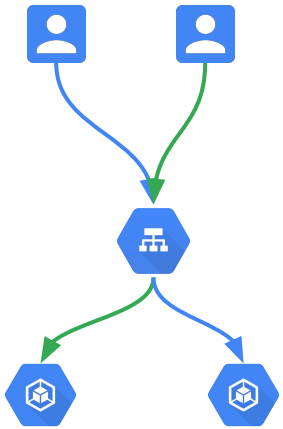
Gateway API Demo 1:
Internal HTTP LB: 跨名稱空間+藍綠部署



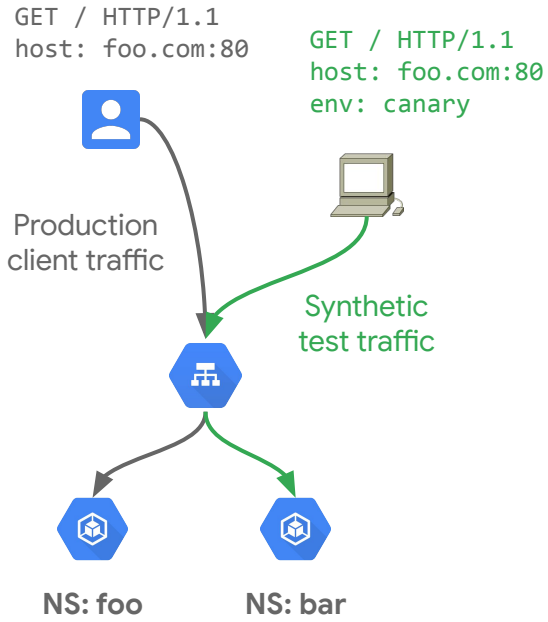
Google Cloud



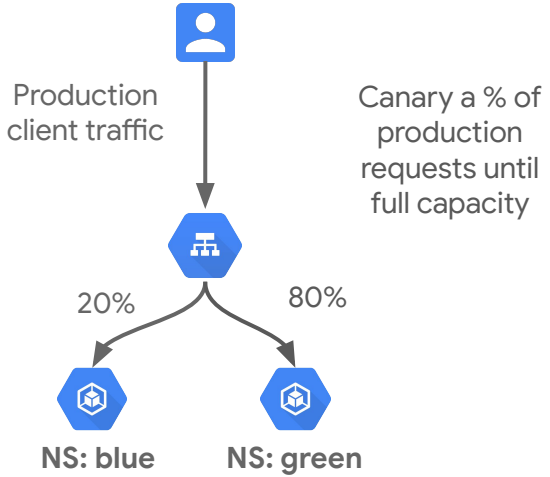
使用情境:



FQDN/憑證控制



Cross Namespace AB Test

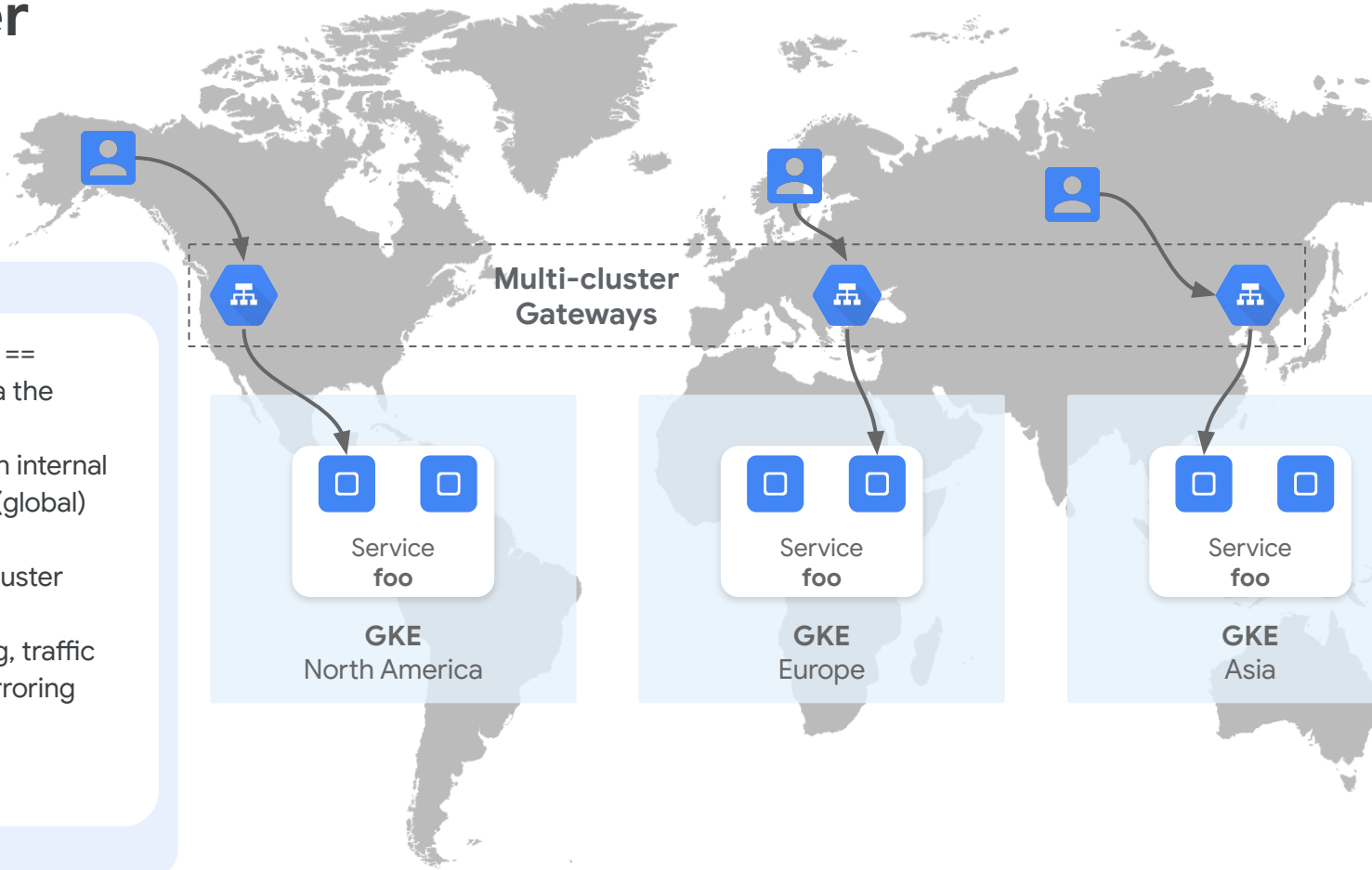


金絲雀部署

MultiCluster GatewayAPI + Istio:
StormBreaker + Thor Hammer

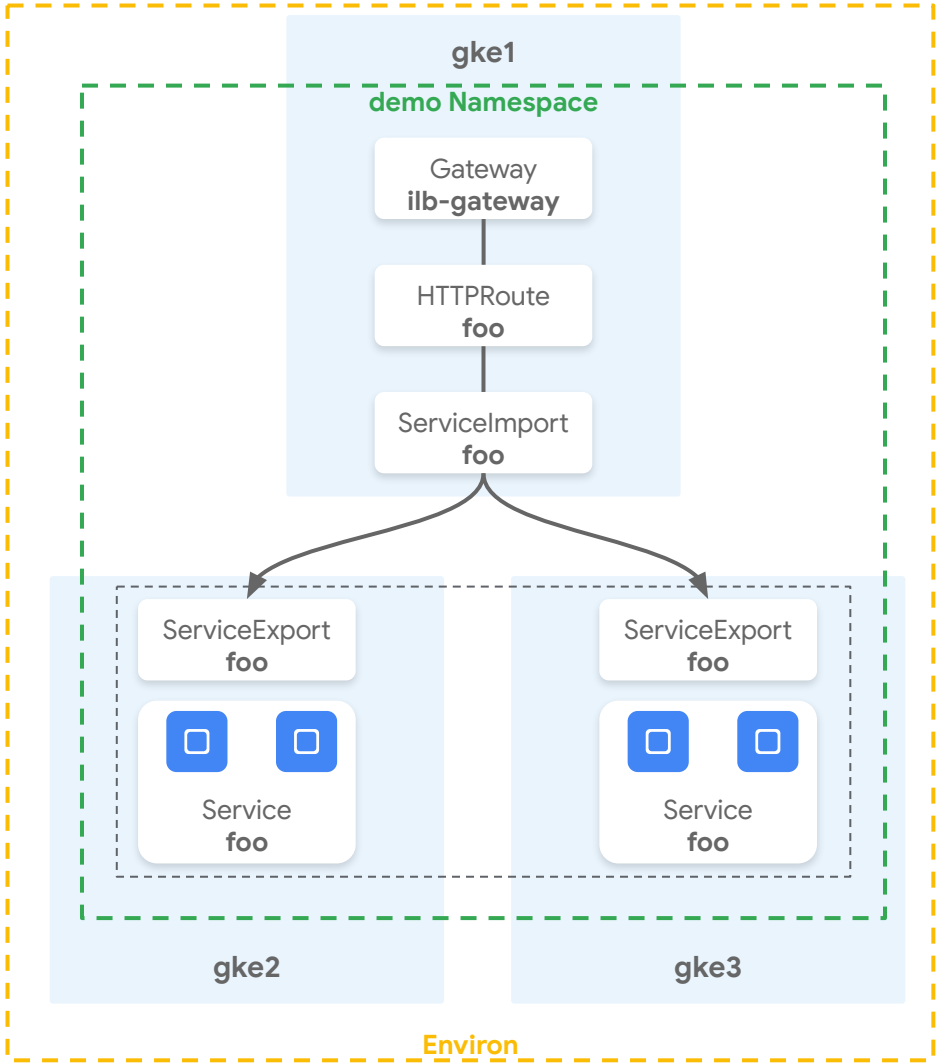
Multi-Cluster Gateways

- Multi-cluster Gateways == Multi-cluster Ingress via the Gateway API
- Now supported for both internal (regional) and external (global) load balancing
- Integrated with Multi-cluster Services (MCS)
- Supports traffic splitting, traffic capacity, and traffic mirroring between clusters



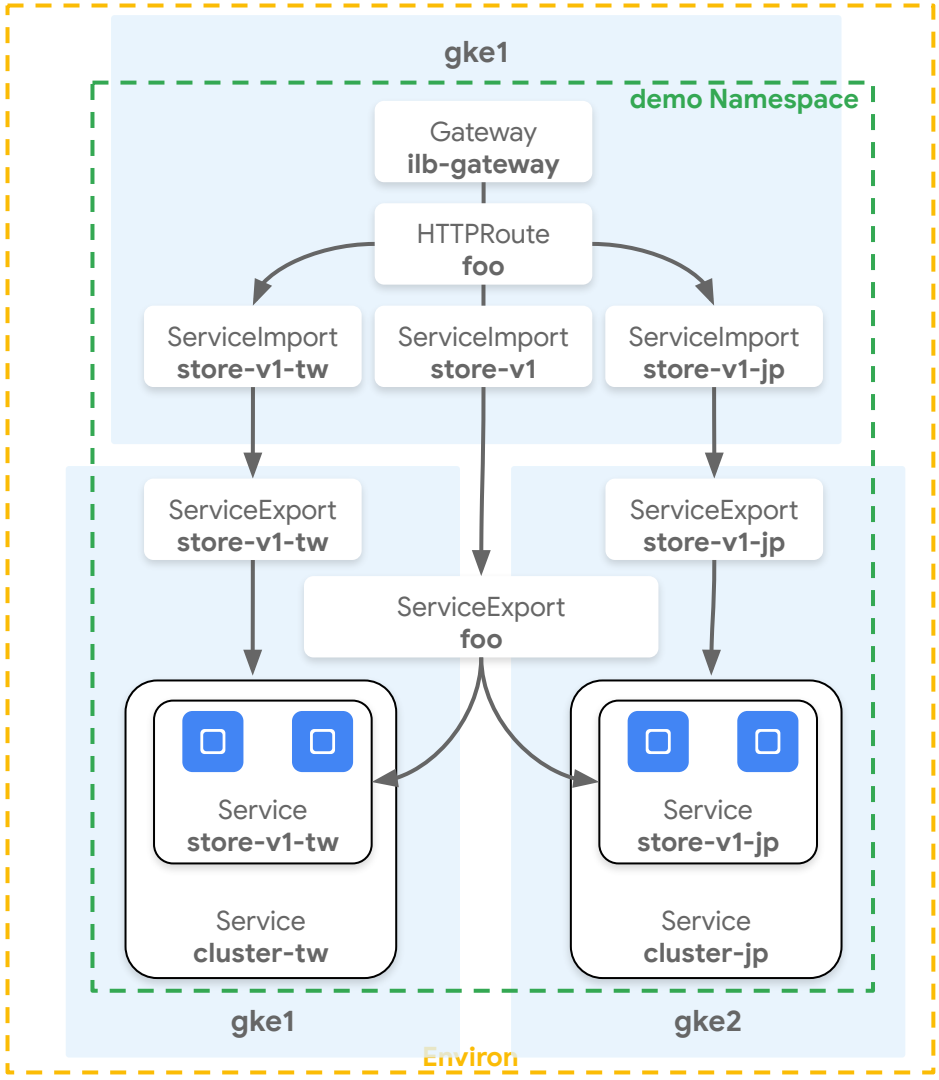
Multi-Cluster Gateways
+
Multi-cluster Services

- MCS introduces two APIs that allows users to create multi-cluster Services
  - **ServiceExport** maps to a Service and exports the endpoints of that Service across all clusters within the Environ
  - **ServiceImports** represent the aggregation of all those endpoints across clusters for that Service
- An HTTPRoute can reference a Service (for single-cluster load balancing or a ServiceImport (for multi-cluster load balancing

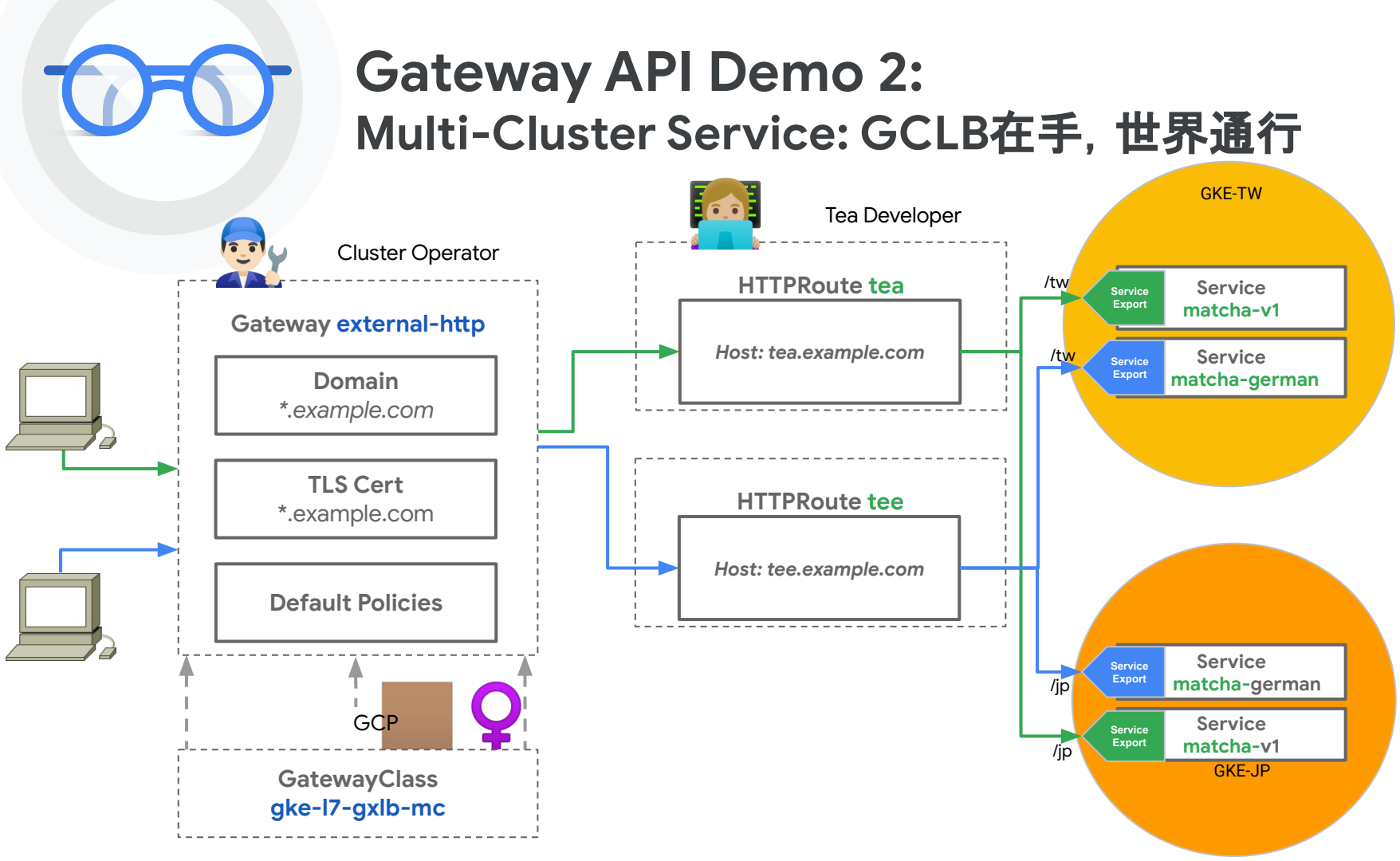


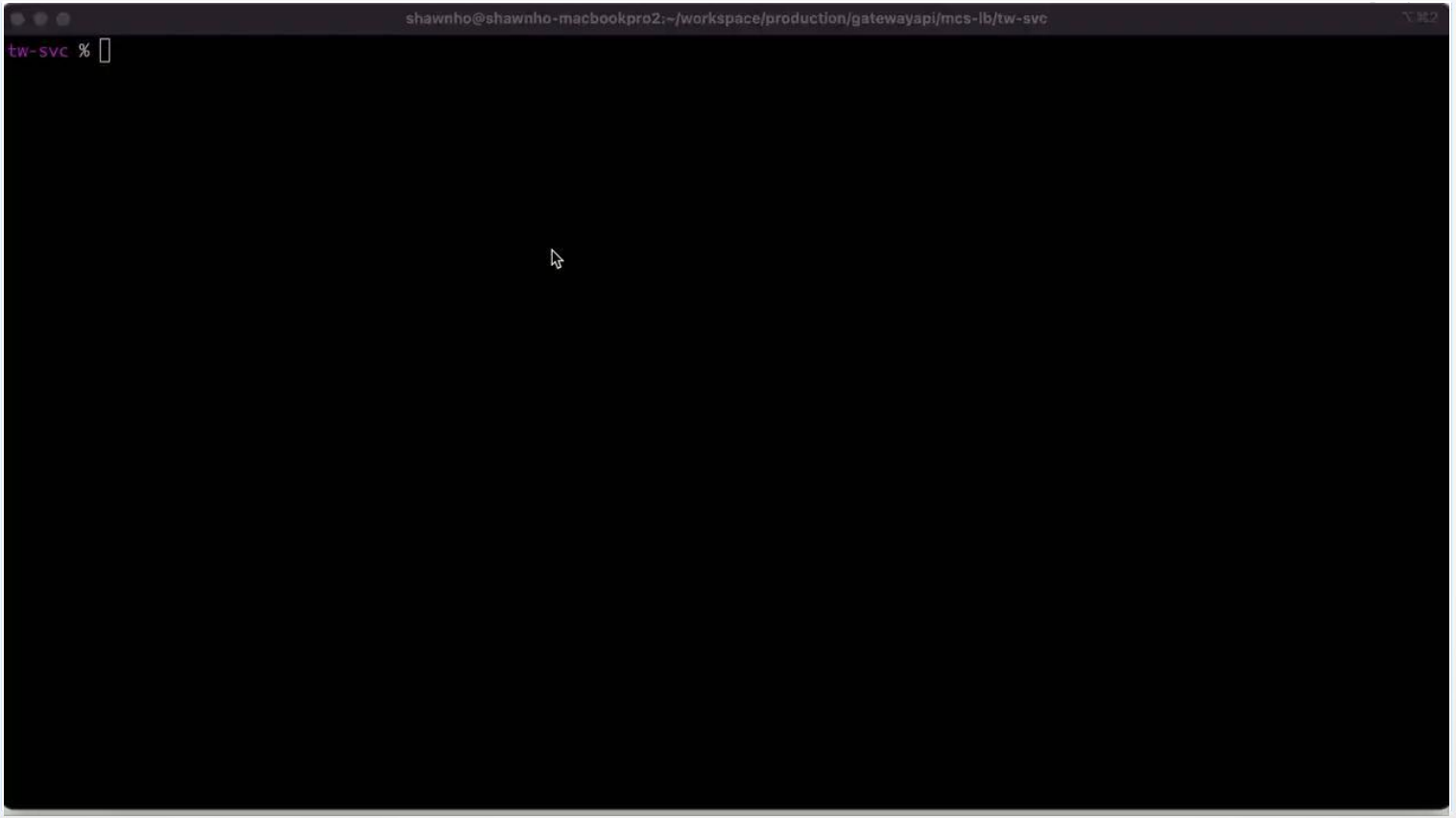
Multi-Cluster Gateways
+
Multi-cluster Services

- Overlapping Services/ServiceExports allow different scopes of endpoints to be targeted by an HTTPRoute
  - foo represents the aggregation of endpoints across gke2/foo and gke3/foo
  - gke2/foo-gke2 and gke/foo-gke3 represent just the foo endpoints in that cluster

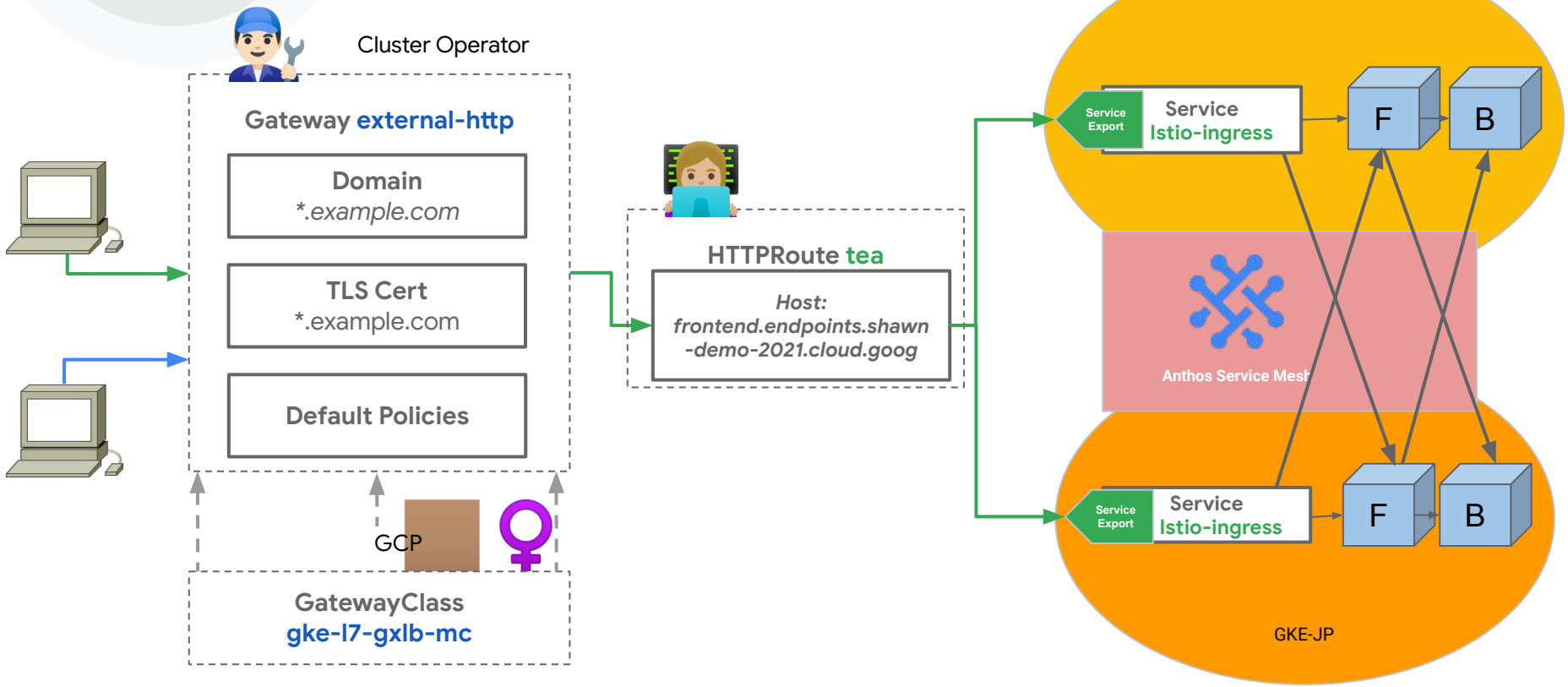


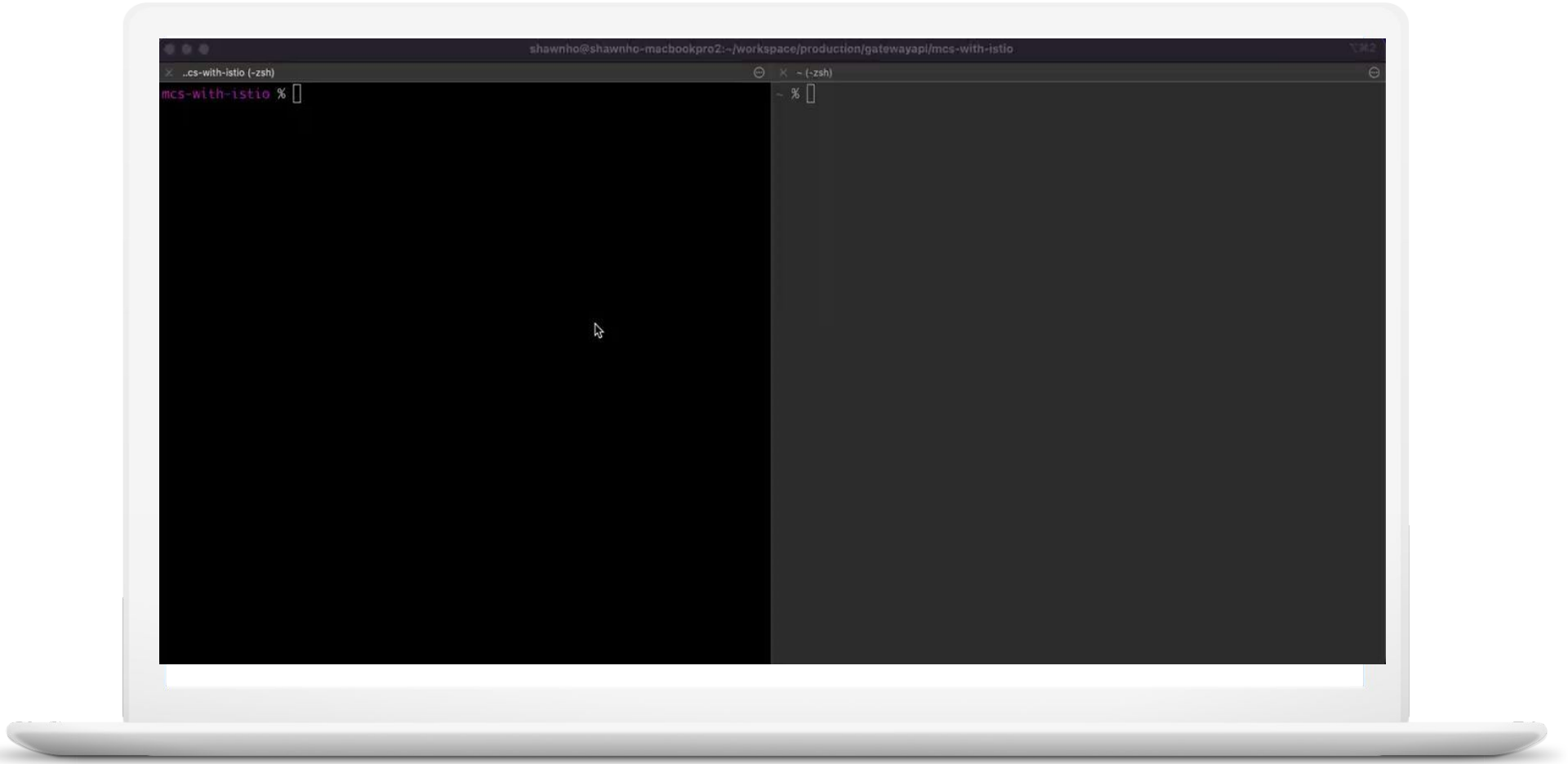
Gateway API Demo 2:
Multi-Cluster Service: GCLB在手，世界通行



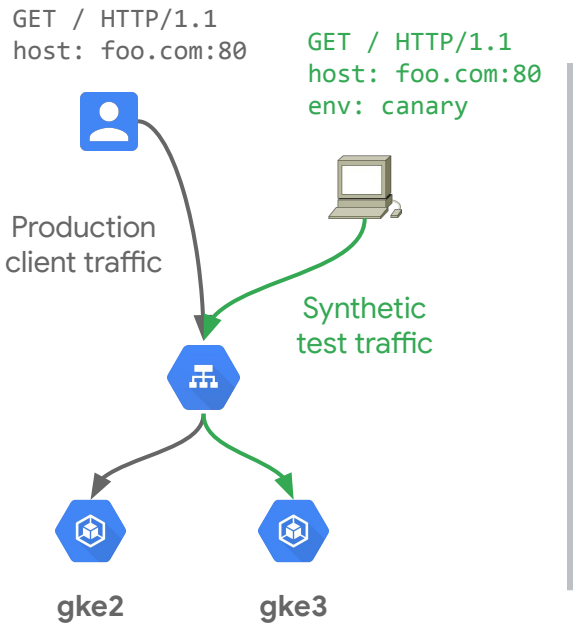


Gateway API Demo 3:
Multi-Cluster Service+Istio: 日不落服務

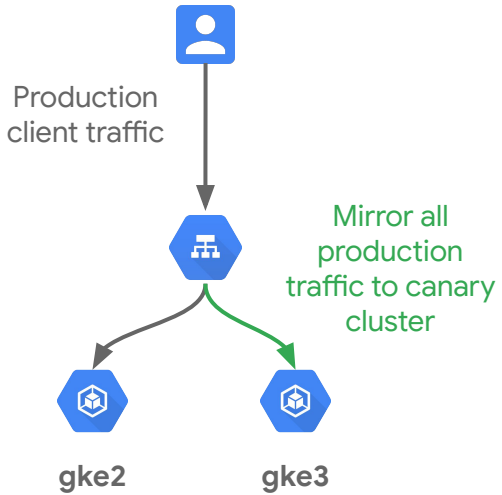



shawnho@shawnho-macbookpro2:~/workspace/production/gatewayapi/mcs-with-istio
..cs-with-istio (-zsh)
mcs-with-istio %
~ (-zsh)
~ %

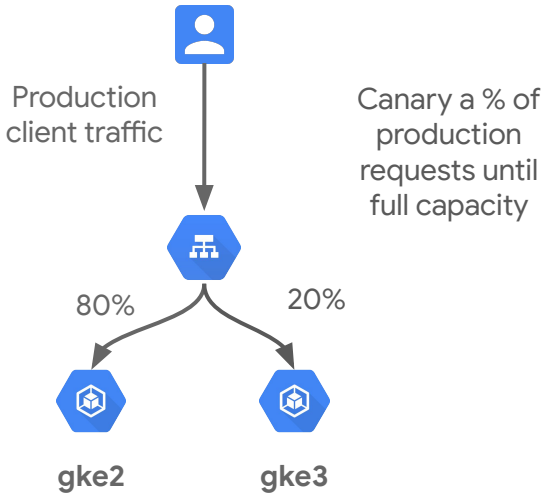
Blue-Green Migration via Multi-cluster Gateways



Synthetic traffic test



Capacity testing with traffic mirroring Or Active-Active



Blue-green traffic splitting

Contour

Using Gateway API v1alpha1 with Contour

Introduction

Gateway API is an open source project managed by the Kubernetes SIG-NETWORK community. The project's goal is to evolve service networking APIs within the Kubernetes ecosystem. Gateway API consists of multiple resources that provide user interfaces to expose Kubernetes applications- Services, Ingress, and more.

This guide covers using version **v1alpha1** of the Gateway API.

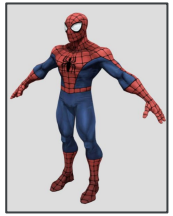
Background

Gateway API targets three personas:

- **Platform Provider**: The Platform Provider is responsible for the overall environment that the cluster runs in, i.e. the cloud provider. The Platform Provider will interact with GatewayClass and Contour resources.
- **Platform Operator**: The Platform Operator is responsible for overall cluster administration. They manage policies, network access, application permissions and will interact with the Gateway resource.
- **Service Operator**: The Service Operator is responsible for defining application configuration and service composition. They will interact with xRoute resources and other typical Kubernetes resources.

Gateway API contains three primary resources:

- **GatewayClass**: Defines a set of gateways with a common configuration and behavior.
- **Gateway**: Requests a point where traffic can be translated to a Service within the cluster.
- **xRoutes**: Describes how traffic coming via the Gateway maps to the Services.



Ambassador

Gateway API

1 min • read

Using the Gateway API

Ambassador Edge Stack now supports a limited subset of the new `v1alpha1` Gateway API. Note that the Gateway API is not supported when `AMBASSADOR_LEGACY_MODE` is set.

Support is currently limited to the following fields, however this will expand in future releases:

- `Gateway.spec.listeners.port`
- `HTTPRoute.spec.rules.matches.path.type` (`Exact`, `Prefix`, and `RegularExpression`)
- `HTTPRoute.spec.rules.matches.path.value`
- `HTTPRoute.spec.rules.matches.header.type` (`Exact` and `RegularExpression`)
- `HTTPRoute.spec.rules.matches.header.values`
- `HTTPRoute.spec.rules.forwardTo.serviceName`
- `HTTPRoute.spec.rules.forwardTo.port`
- `HTTPRoute.spec.rules.forwardTo.weight`

Please see the specification for more details.

Istio

About Blog News Get involved Documentation

Documentation > Tasks > Traffic Management > Ingress > Kubernetes Gateway API

Kubernetes Gateway API

5 minute read page test

This task describes how to configure Istio to expose a service outside the service mesh cluster using the Kubernetes Gateway API. These APIs are an actively developed evolution of the Kubernetes Service and Ingress APIs.

This feature is currently considered alpha. Both the API (owned by Kubernetes SIG-NETWORK) and the Istio implementation are likely to change before being promoted further.

Setup

1. The Gateway APIs do not come installed by default on most Kubernetes clusters. Install the Gateway API CRDs if they are not present:

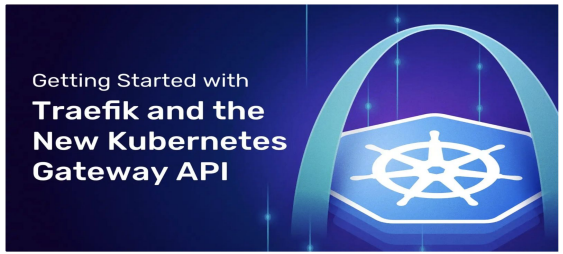
```
$ kubectl get crd gateways.gateway.networking.k8s.io || { kubectl kustomize "github.com/kubernetes-sigs/gateway-api/config/crd?r
```

Differences from Istio APIs

Traefik

Getting Started with Traefik and the New Kubernetes Gateway API

Manuel Zapf • Kubernetes • February 9, 2021

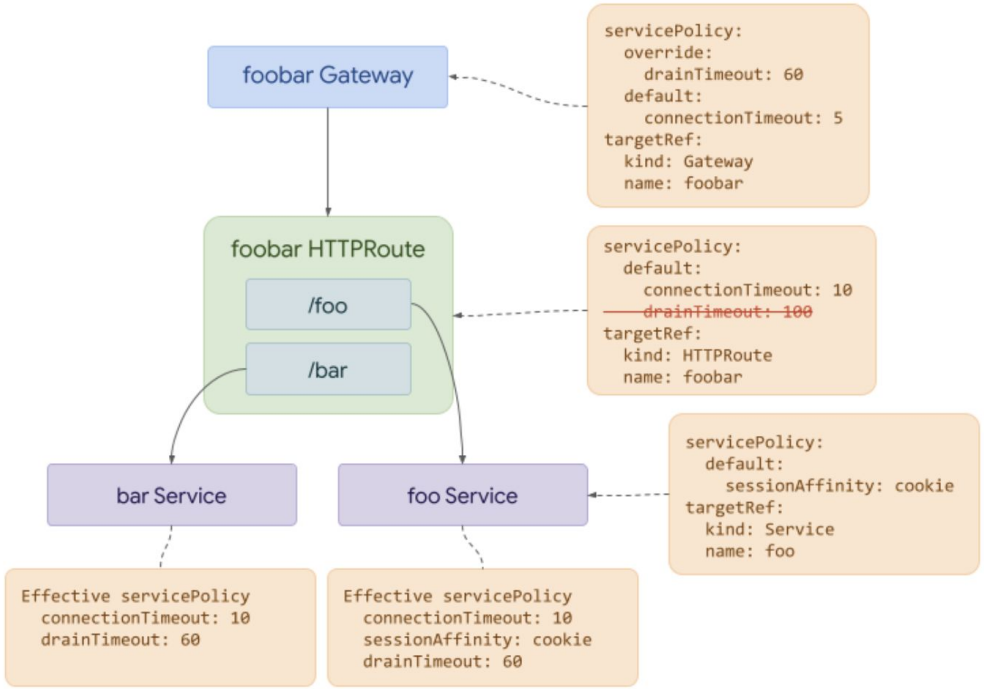


Gateway API Roadmap

External HTTP LoadBalancer with Envoy

Policy API:

- Circuit Break
- Timeout Policy
- Session Persistence Control

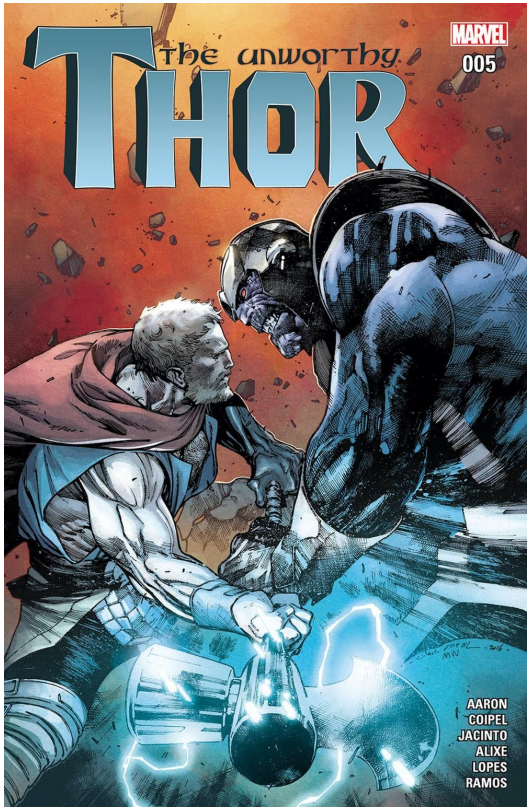


End Game: 3 Take-Aways

GatewayAPI is a multi-tenancy, expandable, kubernetes objects to adapt, which transforms ~~Thor~~ Ingress to be great again.

As GatewayAPI in K8s SIG, it assembles all ~~avengers~~ contributors/vendors into the efforts.

With ~~Captain America~~ Service Mesh's help, we are ready to defeat ~~Thanos~~ Downtime.



Reference

- Gateway API SIG
- GKE Gateway API (Single Cluster) Tutorial
- GKE Gateway API (MultiCluster) Tutorial

Q&A