# Weighted routing in Kubernetes using Istio

Pavan Kumar
Jul 24, 2020 · 4 min read

Let us suppose version:v1 of your application is being used by your customers. A new version:v2 of an application is developed and it has to be deployed in the production environment. But we want to enable only a few customers in the version:v2 and then completely migrate from version:v1 to version:v2 without downtime. In this article let's learn how we can enable weighted routing in our Kubernetes cluster using Istio.
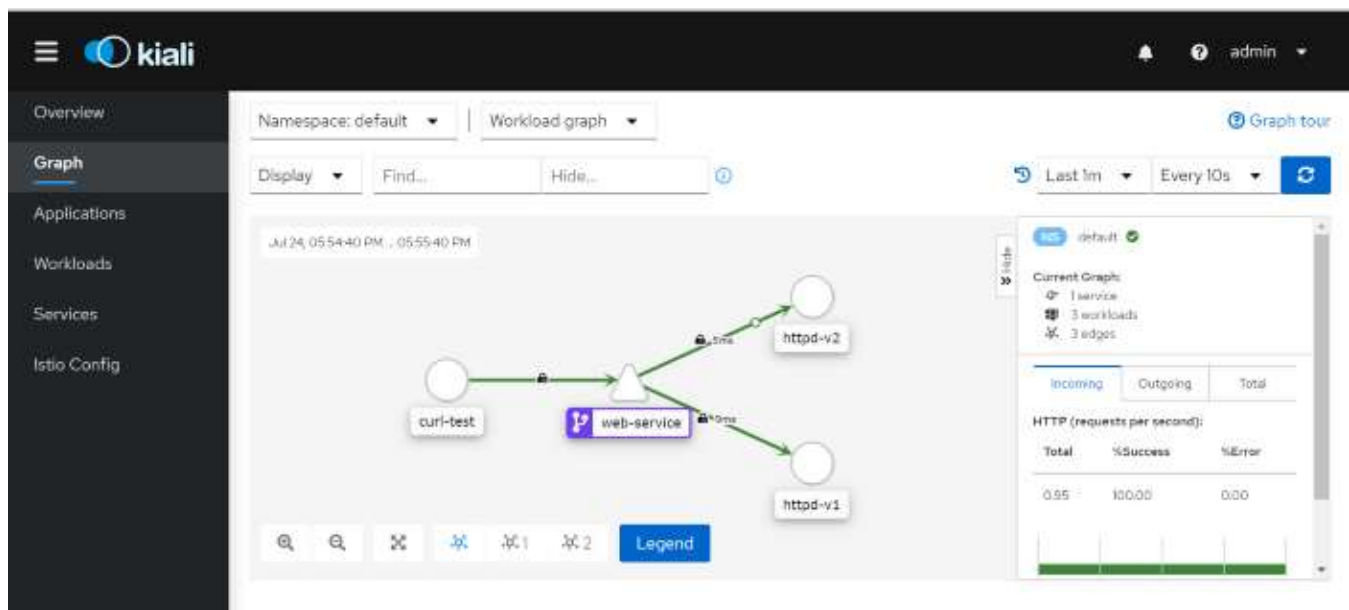
Kiali Dashboard showing the weighted routing between version v1&v2

Before you decide to enable the features of Istio make sure that you label your namespace with the following label **istio-injection=enabled.** When you label the namespace the injection webhook is enabled and any new pods created in that namespace will automatically have the sidecar proxy (envoy proxy) enabled.

Now Deploy the Pod having the webpage serving the version:v1 contents

```yaml
 1  apiVersion: v1
 2  kind: Pod
 3  metadata:
 4    labels:
 5      app: httpd
 6      version: v1
 7    name: httpd-v1
 8    namespace: default
 9  spec:
10    containers:
11    - image: httpd
12      name: httpd
13      resources: {}
14      volumeMounts:
15      - mountPath: /usr/local/apache2/htdocs
16        name: index-html
17    dnsPolicy: ClusterFirst
18    initContainers:
19    - command:
20      - sh
21      - -c
22      - mkdir /usr/local/apache2/htdocs;( echo '<html> <body> <h1>This is version V1!</h1>
23        </body></html>' ) > /usr/local/apache2/htdocs/index.html
24      image: busybox
25      name: busybox
26      volumeMounts:
27      - mountPath: /usr/local/apache2/htdocs
28        name: index-html
29    restartPolicy: Always
30    volumes:
31    - emptyDir: {}
32      name: index-html
```

podv1.yaml hosted with ♡ by **GitHub**   view raw

Expose your pod via ClusterIP service as we would be testing the weighted rules using a curl command inside the cluster. You can expose the service as a LoadBalancer and test it in the browser as well. Since I have Installed my Kubernetes cluster in Oracle Virtual Box I would be exposing the pod via ClusterIP and use the curl command to test the weighted rules.

```yaml
 1  apiVersion: v1
 2  kind: Service
 3  metadata:
 4    labels:
 5      app: web-service
```

```
 6      name: web-service
 7    spec:
 8      ports:
 9      - name: http-port
10        port: 80
11        protocol: TCP
12        targetPort: 80
13      selector:
14        app: httpd
15      type: ClusterIP
```

web-service.yaml hosted with ♡ by GitHub   view raw

Now the requests are only sent to version:v1 when the curl command is executed

[root@master istio-medium]# kubectl run curl-test — image=odise/busybox-curl —
rm -it — /bin/sh -c "while true; do curl web-service; sleep 1; done"
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>
<html> <body> <h1>This is version V1!</h1> </body></html>

Let us configure the Virtual service and Destination Rule to configure the weight of traffic

```
 1    apiVersion: networking.istio.io/v1beta1
 2    kind: VirtualService
 3    metadata:
 4      name: web-virtualservice
 5    spec:
 6      hosts:
 7      - web-service.default.svc.cluster.local
 8      http:
 9      - route:
10        - destination:
11            host: web-service.default.svc.cluster.local
12            subset: v2
13          weight: 25
14        - destination:
15            host: web-service.default.svc.cluster.local
16            subset: v1
17          weight: 75
```

```
17          weight: 75
18
19
20    ---
21
22    apiVersion: networking.istio.io/v1beta1
23    kind: DestinationRule
24    metadata:
25      name: web-destinationrule
26    spec:
27      host: web-service.default.svc.cluster.local
28      subsets:
29      - name: v1
30        labels:
31          version: v1
32      - name: v2
33        labels:
34          version: v2
```

virtual-service.yaml hosted with ♡ by **GitHub**   view raw

Let us try to understand what is the above file doing. When Istio is Installed in your Kubernetes Cluster a lot of CRD's are deployed. The CRD's that we would be focusing on now is Virtual Service and Destination Rule. A Virtual Service defines a set of traffic routing rules for Kubernetes service. And Destination Rules specify where the traffic should be routed once the routing rules are met. Let us examine the YAML file in detail.

Virtual Service:

a) name: Specifies the name of the Virtual Service

b) hosts: The host to which traffic is sent. Here the host is the DNS name of our Kubernetes Service

c) http: It is the list of routing rules for HTTP traffic

d) subset: The name of the subset that the traffic should be directed to which is defined in the corresponding Destination Rule

e) weight: The amount of traffic to be forwarded to the service version

Destination Rule:

a) name: The name of the Destination Rule

b) hosts: The host to which traffic is sent. Here the host is the DNS name of our Kubernetes Service

c) subsets: Named set that represents the Individual version of the service

d) subsets.name: Name of the subset

e) labels: Map of the labels that are used to select the pods

```
[root@master istio-medium]# k get vs
NAME                    GATEWAYS    HOSTS                                      AGE
web-virtualservice                  [web-service.default.svc.cluster.local]   3s
[root@master istio-medium]# k get dr
NAME                    HOST                                      AGE
web-destinationrule     web-service.default.svc.cluster.local     6s
[root@master istio-medium]#
```

Virtual Service and Destination Rule

Now deploy a new pod with the new version:v2. The pod would be picked by the web-service service selector.

```
1   apiVersion: v1
2   kind: Pod
3   metadata:
4     labels:
5       app: httpd
6       version: v2
7     name: httpd-v2
8     namespace: default
9   spec:
10    containers:
11    - image: httpd
12      name: httpd
13      resources: {}
14      volumeMounts:
15      - mountPath: /usr/local/apache2/htdocs
16        name: index-html
17    dnsPolicy: ClusterFirst
18    initContainers:
19    - command:
20      - sh
21      - -c
22      - "mkdir /usr/local/apache2/htdocs;(
23   echo '<html>
24   <body>
25   <h1>This is version V2!</h1>
```

```
26   </body></html>'
27   ) > /usr/local/apache2/htdocs/index.html"
28      name: busybox
29      image: busybox
30      volumeMounts:
31      - mountPath: /usr/local/apache2/htdocs
32        name: index-html
33    restartPolicy: Always
34    volumes:
35    - emptyDir: {}
36      name: index-html
```

podv2.yaml hosted with ♡ by GitHub   view raw

As the virtual service and Destination rules are deployed the magic of Istio is seen now. You can see that 75% percent of the traffic is sent to version:v1 but only 25% of the traffic is sent to version:v2.

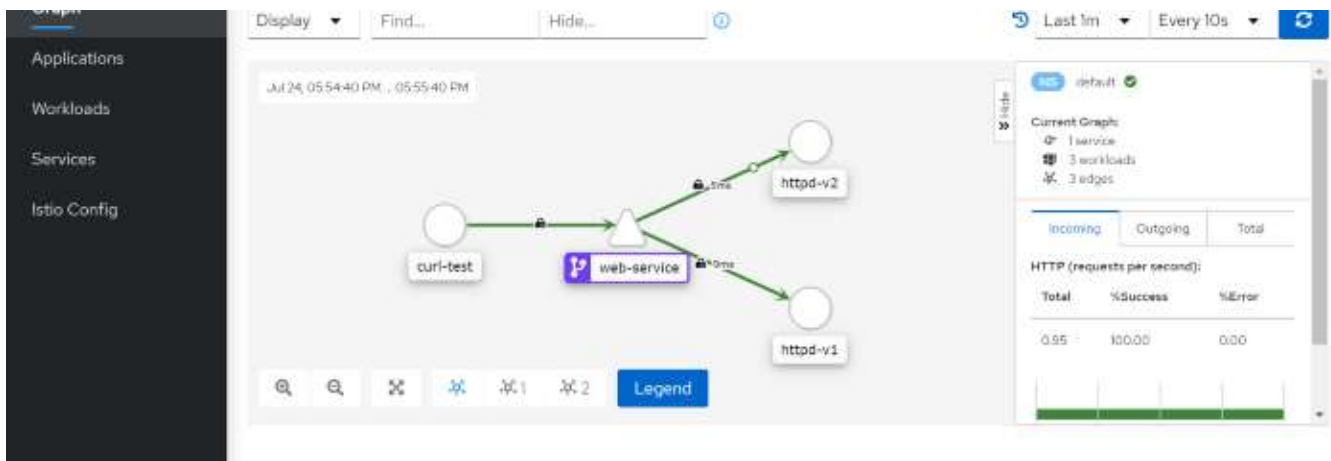This command would launch kiali in your browser.

75% of the traffic to v1 and 25% of the traffic to v2

Incase If you want to visualize the traffic in your closer you can use an istio add-on feature called Kiali. It is a web-based graphical user interface to view service graphs of the mesh and your Istio configuration objects.

> *istioctl dashboard kiali*

Kiali Dashboard

And once the customers are happy with the new version edit the virtual service and change the weight of subset v2 to 100 and subset v1 to 0. You have upgraded your services to the new version with zero downtime :) !!!

Prerequisites

Introduction to Istio Service Mesh

What is Istio Service Mesh?

medium.com

How to Install Istio using istioctl

As discussed in my earlier article we would be working on a few real-time scenarios where we would be able to...

medium.com

Recommended

Mirroring of Live Traffic in Kubernetes using Istio Traffic Mirroring

Why is traffic mirroring needed?

medium.com

MTLS in Istio

Setup Mutual TLS in Istio Service Mesh by plugging in existing CA
Certificates

medium.com

Sign up for Top Weekly Picks

By TechManyu

Tutorials, Solutions, Research articles and cutting-edge techniques delivered once a Week. Make
learning a ritual. Take a look.

Get this newsletter   Emails will be sent to marcus.brito@deal.com.br.
                             Not you?

Istio   Kubernetes   Istio Service Mesh   Service Mesh   Istio Service Tutorial

About   Help   Legal

Get the Medium app