# LAB 6

Sarah Martin

C00257967

Traffic Shifting

Apply weight-based routing

I ran this command to route all traffic to the V1 version and then I opened up the Bookinfo site in my browser. I then

I ran this command to route all traffic to the V1 version:

```
resourceversion:
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-all-v1.yaml

virtualservice.networking.istio.io/productpage created
virtualservice.networking.istio.io/reviews created
virtualservice.networking.istio.io/ratings created
virtualservice.networking.istio.io/details created
```

then I opened up the Bookinfo site in my browser and noted there was no ratings.
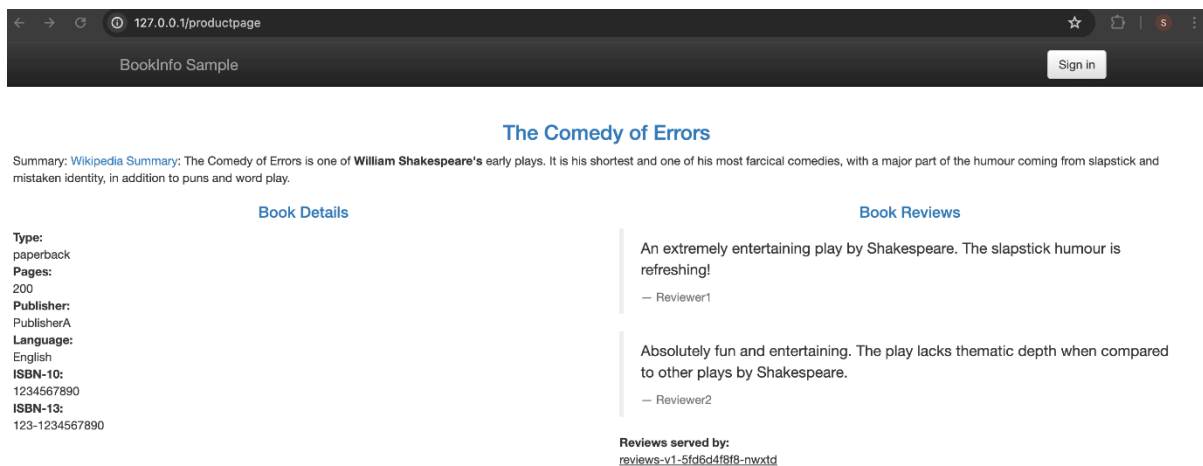
I transferred 50% of the traffic from reviews:v1 to reviews:v3 with the following command. I then waited a few seconds for the new rules to propagate and then confirmed the rule was replaced.

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-50-v3.yaml

virtualservice.networking.istio.io/reviews configured
sarah@Sarahs-MBP istio-1.22.0 % kubectl get virtualservice reviews -o yaml

apiVersion: networking.istio.io/v1
kind: VirtualService
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{},"name":"reviews","namespace":"default"},"spec":{"hosts":["reviews"],"http":[{"route":[{"destination"
:{"host":"reviews","subset":"v1"},"weight":50},{"destination":{"host":"reviews","subset":"v3"},"weight":50}]}]}}
  creationTimestamp: "2024-05-18T13:53:29Z"
  generation: 2
  name: reviews
  namespace: default
  resourceVersion: "1744"
  uid: d5e4186c-a537-45c7-bd2a-25bc9cd2af37
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 50
    - destination:
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-v3.yaml
```

I then refreshed the product page to observe red coloured star ratings approximately 50% of the time.
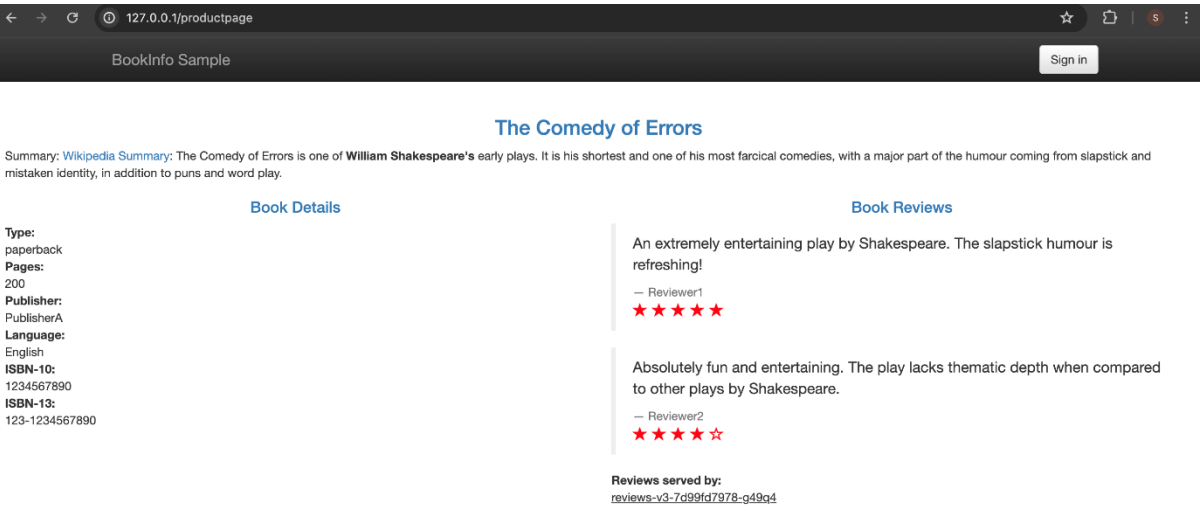
Request Routing

Route to version 1

I ran the following command to create the route rules:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-all-v1.yaml

virtualservice.networking.istio.io/productpage unchanged
virtualservice.networking.istio.io/reviews configured
virtualservice.networking.istio.io/ratings unchanged
virtualservice.networking.istio.io/details unchanged
```

I displayed the defined routes with the following command:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl get virtualservices -o yaml

apiVersion: v1
items:
- apiVersion: networking.istio.io/v1
  kind: VirtualService
  metadata:
    annotations:
      kubectl.kubernetes.io/last-applied-configuration: |
        {"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{},"name":"bookinfo","namespace":"default"},"spec":{"gateways":["bookinfo-gateway"],"hosts":["*"],"ht
tp":[{"match":[{"uri":{"exact":"/productpage"}},{"uri":{"prefix":"/static"}},{"uri":{"exact":"/login"}},{"uri":{"exact":"/logout"}},{"uri":{"prefix":"/api/v1/products"}}],"route":[{"destination":{"host":"
productpage","port":{"number":9080}}}]}]}}
    creationTimestamp: "2024-05-18T13:48:18Z"
    generation: 1
    name: bookinfo
    namespace: default
    resourceVersion: "1212"
    uid: 9f189f90-ba5b-4ea6-a343-9899854519bd
  spec:
    gateways:
    - bookinfo-gateway
    hosts:
    - '*'
    http:
    - match:
      - uri:
          exact: /productpage
      - uri:
          prefix: /static
      - uri:
          exact: /login
      - uri:
          exact: /logout
      - uri:
          prefix: /api/v1/products
      route:
      - destination:
          host: productpage
          port:
            number: 9080
- apiVersion: networking.istio.io/v1
  kind: VirtualService
```

I ran the following command to enable user-based routing:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-reviews-test-v2.yaml

virtualservice.networking.istio.io/reviews configured
sarah@Sarahs-MBP istio-1.22.0 %
```

I confirmed the rule was created:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl get virtualservice reviews -o yaml

apiVersion: networking.istio.io/v1
kind: VirtualService
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{},"name":"reviews","namespace":"default"},"spec":{"hosts":["reviews"],"http":[{"match":[{"headers":{"e
nd-user":{"exact":"jason"}}}],"route":[{"destination":{"host":"reviews","subset":"v2"}}]},{"route":[{"destination":{"host":"reviews","subset":"v1"}}]}]}}
  creationTimestamp: "2024-05-18T13:53:29Z"
  generation: 4
  name: reviews
  namespace: default
  resourceVersion: "2624"
  uid: d5e4186c-a537-45c7-bd2a-25bc9cd2af37
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
sarah@Sarahs-MBP istio-1.22.0 %
```

I logged in as the user jason on the product page and noticed that the star ratings appear next to each review:

BookInfo Sample   👤 jason (sign out)

The Comedy of Errors

Summary: Wikipedia Summary: The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details

Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890

Book Reviews

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!

— Reviewer1

★★★★★

Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.

— Reviewer2

★★★★☆

Reviews served by:
reviews-v2-6f9b55c5db-g28kr

I then logged in as a different user and noticed that the star ratings were gone:

BookInfo Sample   👤 sarah (sign out)

The Comedy of Errors

Summary: Wikipedia Summary: The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details

Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890

Book Reviews

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!

— Reviewer1

Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.

— Reviewer2

Reviews served by:
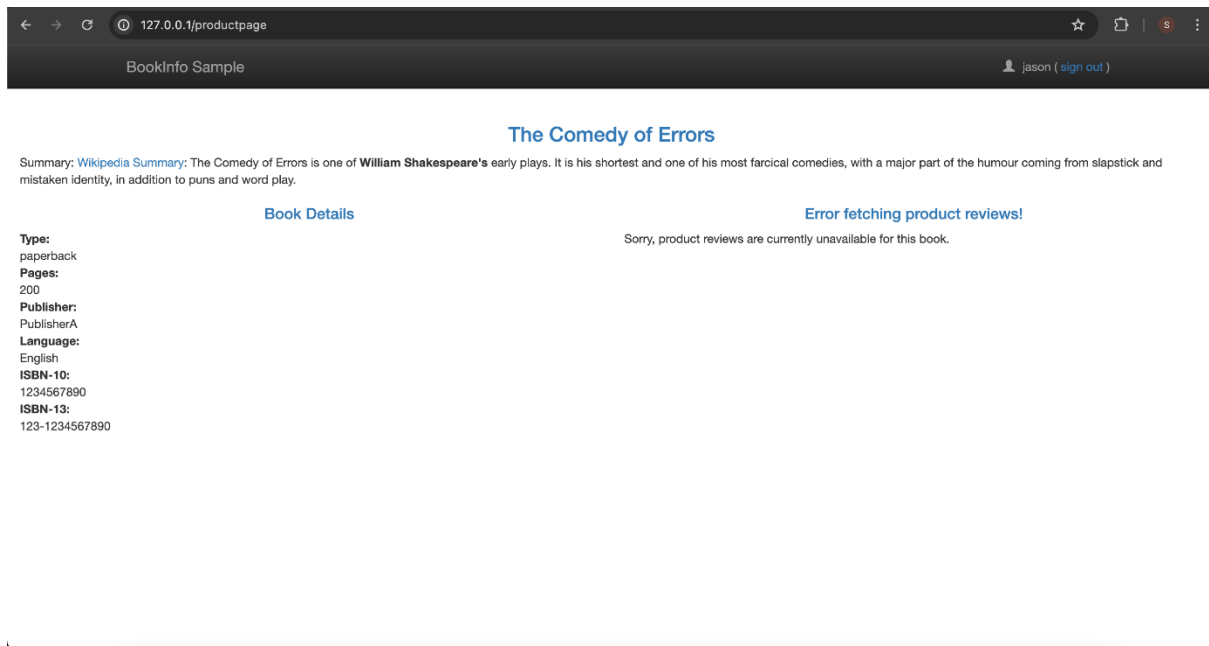reviews-v1-5fd6d4f8f8-nwxtd

Fault Injection

I created a fault injection rule to delay traffic coming from the test user jason

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-ratings-test-delay.yaml

virtualservice.networking.istio.io/ratings configured
sarah@Sarahs-MBP istio-1.22.0 %
```

I confirmed the rule was created

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl get virtualservice ratings -o yaml

apiVersion: networking.istio.io/v1
kind: VirtualService
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{},"name":"ratings","namespace":"default"},"spec":{"hosts":["ratings"],"http":[{"fault":{"delay":{"fixe
dDelay":"7s","percentage":{"value":100}}},"match":[{"headers":{"end-user":{"exact":"jason"}}}],"route":[{"destination":{"host":"ratings","subset":"v1"}}]},{"route":[{"destination":{"host":"ratings","subse
t":"v1"}}]}]}}
  creationTimestamp: "2024-05-18T13:53:29Z"
  generation: 2
  name: ratings
  namespace: default
  resourceVersion: "2945"
  uid: 160fbebb-a6d2-4e12-b7e6-0c11a86d6140
spec:
  hosts:
  - ratings
  http:
  - fault:
      delay:
        fixedDelay: 7s
        percentage:
          value: 100
    match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
sarah@Sarahs-MBP istio-1.22.0 %
```

I opened the Bookinfo web application and logged in as the user jason. I expected the Bookinfo home page to load without errors in approximately 7 seconds. However, there is a problem: the Reviews section displays an error message:

The Comedy of Errors

Summary: Wikipedia Summary: The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details	Error fetching product reviews!
	Sorry, product reviews are currently unavailable for this book.

Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890

I created a fault injection rule to send an HTTP abort for user jason:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f samples/bookinfo/networking/virtual-service-ratings-test-abort.yaml

virtualservice.networking.istio.io/ratings configured
sarah@Sarahs-MBP istio-1.22.0 %
```

I then confirmed the rule was created:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl get virtualservice ratings -o yaml

apiVersion: networking.istio.io/v1
kind: VirtualService
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.istio.io/v1alpha3","kind":"VirtualService","metadata":{"annotations":{},"name":"ratings","namespace":"default"},"spec":{"hosts":["ratings"],"http":[{"fault":{"abort":{"http
Status":500,"percentage":{"value":100}}},"match":[{"headers":{"end-user":{"exact":"jason"}}}],"route":[{"destination":{"host":"ratings","subset":"v1"}}]},{"route":[{"destination":{"host":"ratings","subset
":"v1"}}]}]}}
  creationTimestamp: "2024-05-18T13:53:29Z"
  generation: 3
  name: ratings
  namespace: default
  resourceVersion: "3191"
  uid: 160fbebb-a6d2-4e12-b7e6-0c11a86d6140
spec:
  hosts:
  - ratings
  http:
  - fault:
      abort:
        httpStatus: 500
        percentage:
          value: 100
    match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: ratings
        subset: v1
  - route:
    - destination:
        host: ratings
        subset: v1
sarah@Sarahs-MBP istio-1.22.0 %
```

Testing the abort configuration

I opened the Bookinfo web application and logged in as the user jason. The rating services is currently unavailable message appears:
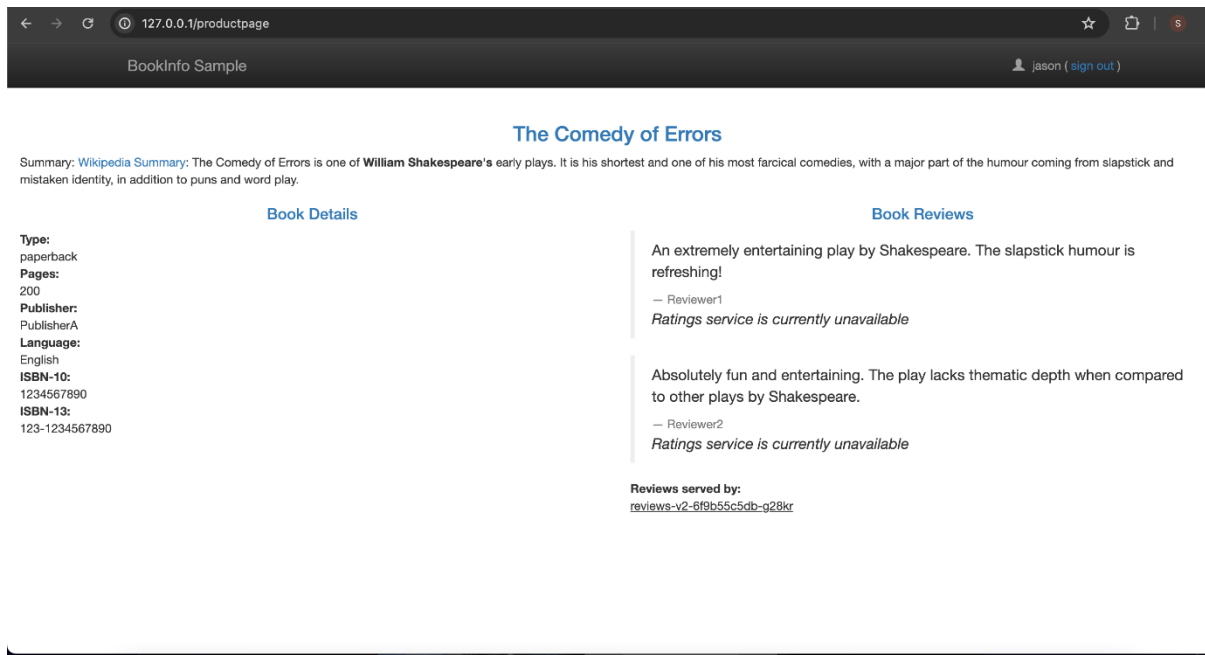
I then logged out as jason and noticed that /productpage still calls reviews:v1 for everybody but jason. I did not see any error message:
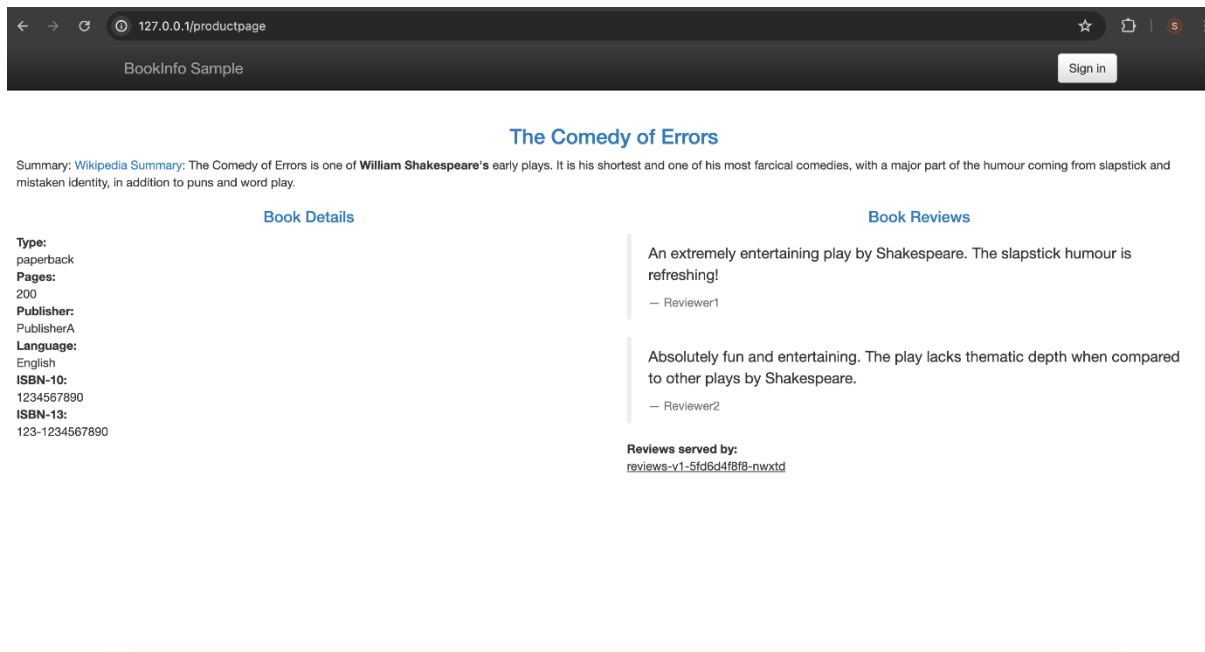
Circuit Breaking

Configuring the circuit break

I created a destination rule to apply circuit breaking settings when calling the httpbin service:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f - <<EOF
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: httpbin
spec:
  host: httpbin
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 1
      http:
        http1MaxPendingRequests: 1
        maxRequestsPerConnection: 1
    outlierDetection:
      consecutive5xxErrors: 1
      interval: 1s
      baseEjectionTime: 3m
      maxEjectionPercent: 100
EOF

destinationrule.networking.istio.io/httpbin created
sarah@Sarahs-MBP istio-1.22.0 %
```

I verified the destination rule was created correctly:

```
destinationrule.networking.istio.io/httpbin created
sarah@Sarahs-MBP istio-1.22.0 % kubectl get destinationrule httpbin -o yaml

apiVersion: networking.istio.io/v1
kind: DestinationRule
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"networking.istio.io/v1alpha3","kind":"DestinationRule","metadata":{"annotations":{},"name":"httpbin","namespace":"default"},"spec":{"host":"httpbin","trafficPolicy":{"connectionPool":
{"http":{"http1MaxPendingRequests":1,"maxRequestsPerConnection":1},"tcp":{"maxConnections":1}},"outlierDetection":{"baseEjectionTime":"3m","consecutive5xxErrors":1,"interval":"1s","maxEjectionPercent":100
}}}}
  creationTimestamp: "2024-05-18T14:16:45Z"
  generation: 1
  name: httpbin
  namespace: default
  resourceVersion: "3494"
  uid: 3f8c5afe-96a4-455b-9c16-c28d6f314be1
spec:
  host: httpbin
  trafficPolicy:
    connectionPool:
      http:
        http1MaxPendingRequests: 1
        maxRequestsPerConnection: 1
      tcp:
        maxConnections: 1
    outlierDetection:
      baseEjectionTime: 3m
      consecutive5xxErrors: 1
      interval: 1s
      maxEjectionPercent: 100
sarah@Sarahs-MBP istio-1.22.0 %
```

Adding a client

I first manually injected the sidecar and then deployed the fortio application:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl apply -f <(istioctl kube-inject -f samples/httpbin/sample-client/fortio-deploy.yaml)

service/fortio created
deployment.apps/fortio-deploy created
sarah@Sarahs-MBP istio-1.22.0 %
```

I logged into the client pod and used the fortio tool to call httpbin. I passed in curl to indicate that I want to make one call:

```
sarah@Sarahs-MBP istio-1.22.0 % export FORTIO_POD=$(kubectl get pods -l app=fortio -o 'jsonpath={.items[0].metadata.name}')
kubectl exec "$FORTIO_POD" -c fortio -- /usr/bin/fortio curl -quiet http://httpbin:8000/get

HTTP/1.1 200 OK
server: envoy
date: Sat, 18 May 2024 14:49:01 GMT
content-type: application/json
content-length: 425
access-control-allow-origin: *
access-control-allow-credentials: true
x-envoy-upstream-service-time: 29

{
  "args": {},
  "headers": {
    "Host": "httpbin:8000",
    "User-Agent": "fortio.org/fortio-1.60.3",
    "X-Envoy-Attempt-Count": "1",
    "X-Forwarded-Client-Cert": "By=spiffe://cluster.local/ns/default/sa/httpbin;Hash=c4eadcaf8970e0d4541f120bfae468063dcc298885ff43ab115e79ae318b2828;Subject=\"\";URI=spiffe://cluster.local/ns/default/sa/
default"
  },
  "origin": "127.0.0.6",
  "url": "http://httpbin:8000/get"
}
sarah@Sarahs-MBP istio-1.22.0 %
```

Tripping the circuit breaker

I called the service with two concurrent connections (-c 2) and send 20 requests (-n 20):

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl exec "$FORTIO_POD" -c fortio -- /usr/bin/fortio load -c 2 -qps 0 -n 20 -loglevel Warning http://httpbin:8000/get

{"ts":1716043800.331615,"level":"info","r":1,"file":"logger.go","line":254,"msg":"Log level is now 3 Warning (was 2 Info)"}
Fortio 1.60.3 running at 0 queries per second, 4->4 procs, for 20 calls: http://httpbin:8000/get
Starting at max qps with 2 thread(s) [gomax 4] for exactly 20 calls (10 per thread + 0)
{"ts":1716043800.344759,"level":"warn","r":11,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043800.371368,"level":"warn","r":12,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043800.378583,"level":"warn","r":11,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043800.382761,"level":"warn","r":12,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043800.387581,"level":"warn","r":11,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043800.392217,"level":"warn","r":12,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043800.400721,"level":"warn","r":11,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
Ended after 73.706429ms : 20 calls. qps=271.35
Aggregated Function Time : count 20 avg 0.0072830049 +/- 0.005075 min 0.000432662 max 0.01795933 sum 0.145660098
# range, mid point, percentile, count
>= 0.000432662 <= 0.001 , 0.000716331 , 20.00, 4
> 0.001 <= 0.002 , 0.0015 , 25.00, 1
> 0.004 <= 0.005 , 0.0045 , 35.00, 2
> 0.005 <= 0.006 , 0.0055 , 40.00, 1
> 0.006 <= 0.007 , 0.0065 , 50.00, 2
> 0.008 <= 0.009 , 0.0085 , 70.00, 4
> 0.009 <= 0.01 , 0.0095 , 75.00, 1
> 0.01 <= 0.011 , 0.0105 , 80.00, 1
> 0.012 <= 0.014 , 0.013 , 90.00, 2
> 0.016 <= 0.0179593 , 0.0169797 , 100.00, 2
# target 50% 0.007
# target 75% 0.01
# target 90% 0.014
# target 99% 0.0177634
# target 99.9% 0.0179397
Error cases : count 7 avg 0.002731214 +/- 0.003292 min 0.000432662 max 0.00950636 sum 0.019118498
# range, mid point, percentile, count
>= 0.000432662 <= 0.001 , 0.000716331 , 57.14, 4
> 0.001 <= 0.002 , 0.0015 , 71.43, 1
> 0.005 <= 0.006 , 0.0055 , 85.71, 1
> 0.009 <= 0.00950636 , 0.00925318 , 100.00, 1
# target 50% 0.000905444
# target 75% 0.00525
# target 90% 0.00915191
# target 99% 0.00947091
# target 99.9% 0.00950282
# Socket and IP used for each connection:
[0]   5 socket used, resolved to 10.102.40.199:8000, connection timing : count 5 avg 0.000286651 +/- 0.0003104 min 0.000101138 max 0.000905592 sum 0.001433255
[1]   4 socket used, resolved to 10.102.40.199:8000, connection timing : count 4 avg 0.000567066 +/- 0.000211 min 0.000356108 max 0.000903246 sum 0.002268264
Connection time (s) : count 9 avg 0.00041127989 +/- 0.0003045 min 0.000101138 max 0.000905592 sum 0.003701519
Sockets used: 9 (for perfect keepalive, would be 2)
Uniform: false, Jitter: false, Catchup allowed: true
IP addresses distribution:
10.102.40.199:8000: 9
Code 200 : 13 (65.0 %)
Code 503 : 7 (35.0 %)
Response Header Sizes : count 20 avg 149.65 +/- 109.8 min 0 max 231 sum 2993
Response Body/Total Sizes : count 20 avg 510.25 +/- 197.6 min 241 max 656 sum 10205
All done 20 calls (plus 0 warmup) 7.283 ms avg, 271.3 qps
sarah@Sarahs-MBP istio-1.22.0 %
```

I then brought the number of concurrent connections up to 3:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl exec "$FORTIO_POD" -c fortio -- /usr/bin/fortio load -c 3 -qps 0 -n 30 -loglevel Warning http://httpbin:8000/get

{"ts":1716043837.755544,"level":"info","r":1,"file":"logger.go","line":254,"msg":"Log level is now 3 Warning (was 2 Info)"}
Fortio 1.60.3 running at 0 queries per second, 4->4 procs, for 30 calls: http://httpbin:8000/get
Starting at max qps with 3 thread(s) [gomax 4] for exactly 30 calls (10 per thread + 0)
{"ts":1716043837.761845,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.764969,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.771317,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.773138,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.778627,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.782826,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.784351,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.785141,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.788235,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.790540,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.791998,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.794077,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.797474,"level":"warn","r":15,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":1,"run":0}
{"ts":1716043837.800194,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.803635,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.806429,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.808522,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.813055,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.815989,"level":"warn","r":16,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":2,"run":0}
{"ts":1716043837.822076,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
{"ts":1716043837.825464,"level":"warn","r":14,"file":"http_client.go","line":1104,"msg":"Non ok http code","code":503,"status":"HTTP/1.1 503","thread":0,"run":0}
Ended after 69.398798ms : 30 calls. qps=432.28
Aggregated Function Time : count 30 avg 0.0065590035 +/- 0.005676 min 0.000710301 max 0.02018483 sum 0.196770105
# range, mid point, percentile, count
>= 0.000710301 <= 0.001 , 0.000855151 , 10.00, 3
> 0.001 <= 0.002 , 0.0015 , 13.33, 1
> 0.002 <= 0.003 , 0.0025 , 33.33, 6
> 0.003 <= 0.004 , 0.0035 , 53.33, 6
> 0.004 <= 0.005 , 0.0045 , 56.67, 1
> 0.005 <= 0.006 , 0.0055 , 66.67, 3
> 0.006 <= 0.007 , 0.0065 , 70.00, 1
> 0.011 <= 0.012 , 0.0115 , 76.67, 2
> 0.012 <= 0.014 , 0.013 , 83.33, 2
> 0.014 <= 0.016 , 0.015 , 90.00, 2
> 0.016 <= 0.018 , 0.017 , 93.33, 1
> 0.018 <= 0.02 , 0.019 , 96.67, 1
> 0.02 <= 0.0201848 , 0.0200924 , 100.00, 1
# target 50% 0.00383333
# target 75% 0.01175
# target 90% 0.016
# target 99% 0.0201294
# target 99.9% 0.0201793
Error cases : count 21 avg 0.0031045628 +/- 0.001582 min 0.000710301 max 0.006233534 sum 0.065195819
# range, mid point, percentile, count
>= 0.000710301 <= 0.001 , 0.000855151 , 14.29, 3
> 0.001 <= 0.002 , 0.0015 , 19.05, 1
> 0.002 <= 0.003 , 0.0025 , 47.62, 6
> 0.003 <= 0.004 , 0.0035 , 76.19, 6
> 0.004 <= 0.005 , 0.0045 , 80.95, 1
> 0.005 <= 0.006 , 0.0055 , 95.24, 3
> 0.006 <= 0.00623353 , 0.00611677 , 100.00, 1
# target 50% 0.00308333
# target 75% 0.00395833
```

I queried the istio-proxy stats to see more:

```
sarah@Sarahs-MBP istio-1.22.0 % kubectl exec "$FORTIO_POD" -c istio-proxy -- pilot-agent request GET stats | grep httpbin | grep pending

cluster.outbound|8000||httpbin.default.svc.cluster.local.circuit_breakers.default.remaining_pending: 1
cluster.outbound|8000||httpbin.default.svc.cluster.local.circuit_breakers.default.rq_pending_open: 0
cluster.outbound|8000||httpbin.default.svc.cluster.local.circuit_breakers.high.rq_pending_open: 0
cluster.outbound|8000||httpbin.default.svc.cluster.local.upstream_rq_pending_active: 0
cluster.outbound|8000||httpbin.default.svc.cluster.local.upstream_rq_pending_failure_eject: 0
cluster.outbound|8000||httpbin.default.svc.cluster.local.upstream_rq_pending_overflow: 28
cluster.outbound|8000||httpbin.default.svc.cluster.local.upstream_rq_pending_total: 23
sarah@Sarahs-MBP istio-1.22.0 %
```