Abstract

cloud native, need for resiliency, service meshes as solution, demo, pros of istio – resiliency and fault injection

Evaluation and Discussion

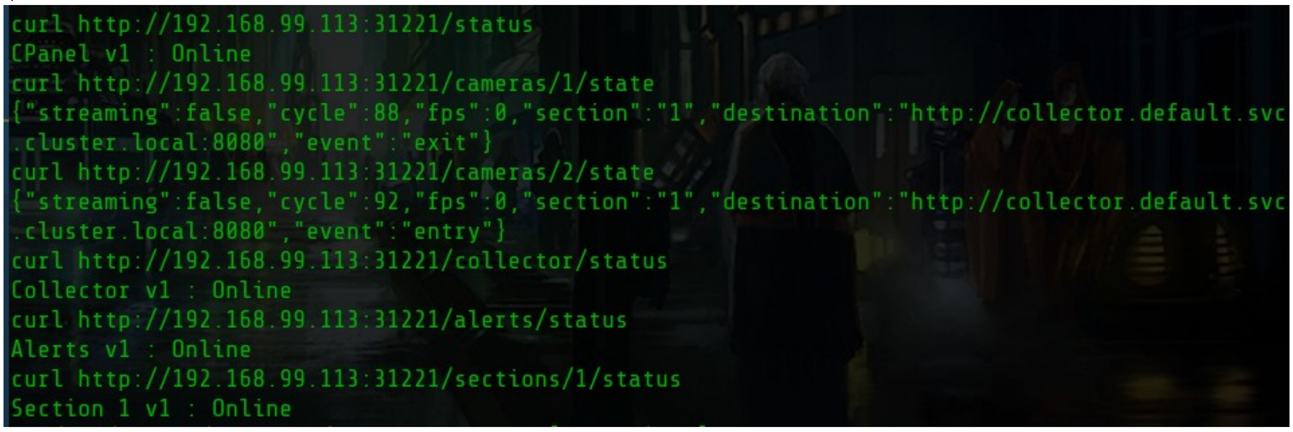
- demo works, tests, screenshots/results
- comparison to k8s only

Kubernetes has only round robin load balancing. Istio with the help of destinations rules extends native kubernetes load balancing and presents the following types: random, round robin, weighted least request, ring hash (#istio). In such a case istio can give any microservice replica set it's own load balancer. To show how istio load balancing can be configured, we need first to learn about routing mechanism provided by istio.

Istio routing mechanism

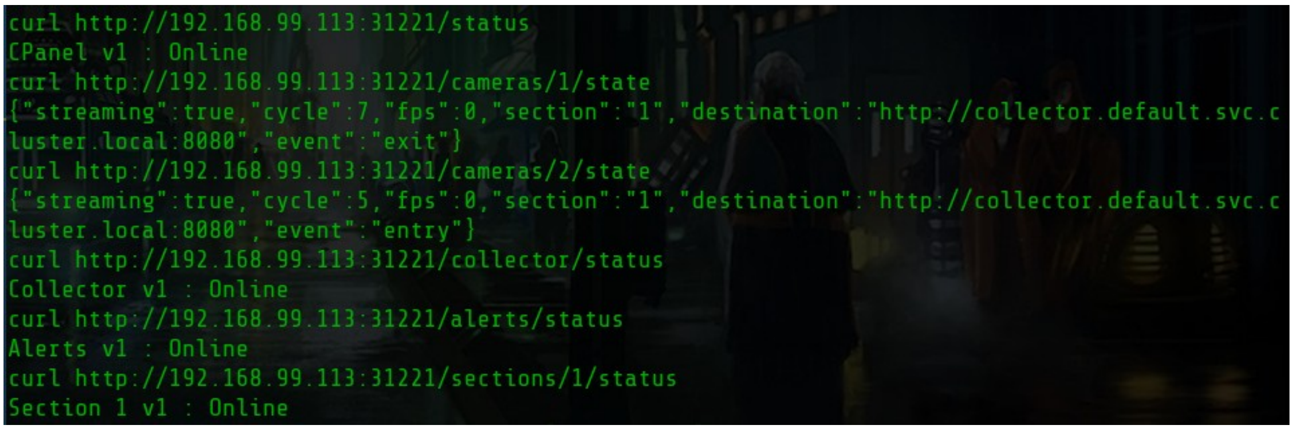
This solution can be used to make canary deployments and also make user experience more resilient - "user resilience". For example, new version of service can be made available only to one group of users (test group). It can be as much as only 1% of of the hole traffic. Users can be filtered by headers in http request. If something goes wrong with new version of service it is very easy to rollback and switch all the traffic back to production version.
This mechanism allows also to do blue/green deployments.

$ make deploy-app-default
$ make deploy-istio-default
$ make health



```
curl http://192.168.99.113:31221/status
CPanel v1 : Online
curl http://192.168.99.113:31221/cameras/1/state
{"streaming":false,"cycle":88,"fps":0,"section":"1","destination":"http://collector.default.svc
.cluster.local:8080","event":"exit"}
curl http://192.168.99.113:31221/cameras/2/state
{"streaming":false,"cycle":92,"fps":0,"section":"1","destination":"http://collector.default.svc
.cluster.local:8080","event":"entry"}
curl http://192.168.99.113:31221/collector/status
Collector v1 : Online
curl http://192.168.99.113:31221/alerts/status
Alerts v1 : Online
curl http://192.168.99.113:31221/sections/1/status
Section 1 v1 : Online
```

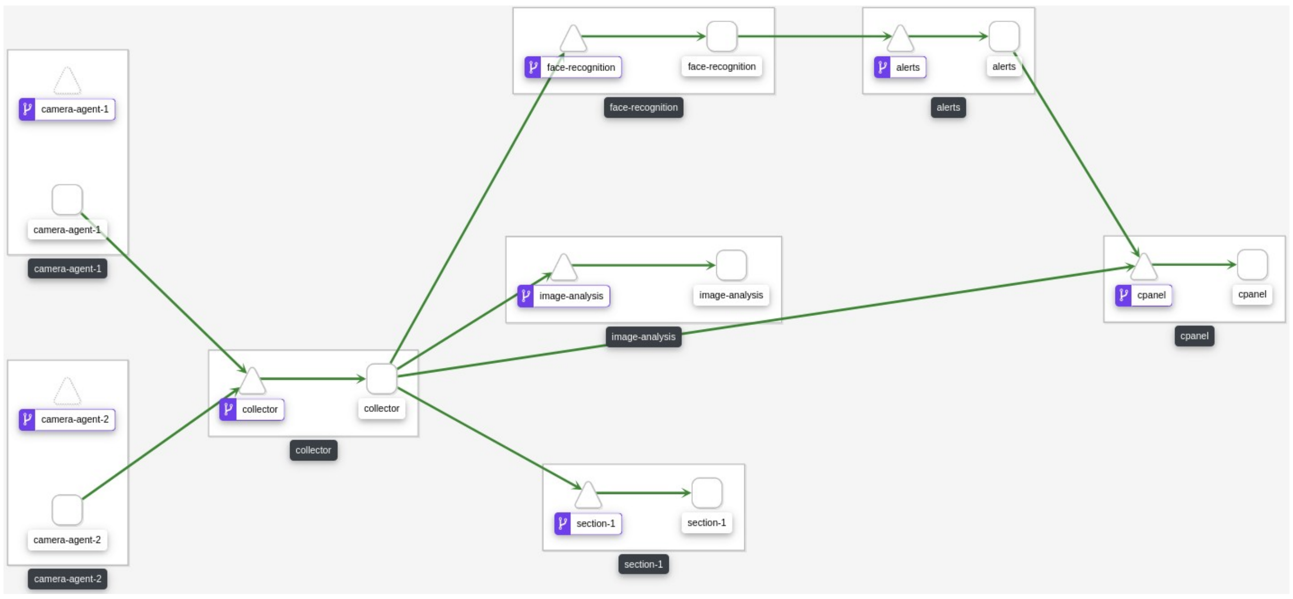
$ make start-cameras
$ make health

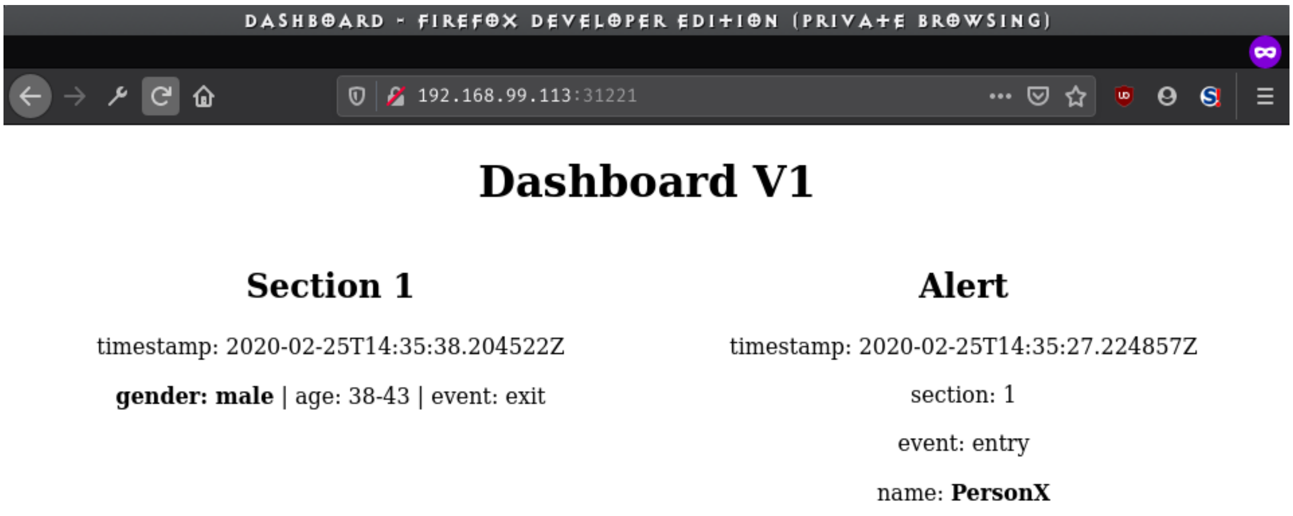


```
curl http://192.168.99.113:31221/status
CPanel v1 : Online
curl http://192.168.99.113:31221/cameras/1/state
{"streaming":true,"cycle":7,"fps":0,"section":"1","destination":"http://collector.default.svc.c
luster.local:8080","event":"exit"}
curl http://192.168.99.113:31221/cameras/2/state
{"streaming":true,"cycle":5,"fps":0,"section":"1","destination":"http://collector.default.svc.c
luster.local:8080","event":"entry"}
curl http://192.168.99.113:31221/collector/status
Collector v1 : Online
curl http://192.168.99.113:31221/alerts/status
Alerts v1 : Online
curl http://192.168.99.113:31221/sections/1/status
Section 1 v1 : Online
```

curl http://192.168.99.113:31221/status
CPanel v1 : Online

```
curl http://192.168.99.113:31221/cameras/1/state
{"streaming":true,"cycle":7,"fps":0,"section":"1","destination":"http://
collector.default.svc.cluster.local:8080","event":"exit"}
curl http://192.168.99.113:31221/cameras/2/state
{"streaming":true,"cycle":5,"fps":0,"section":"1","destination":"http://
collector.default.svc.cluster.local:8080","event":"entry"}
curl http://192.168.99.113:31221/collector/status
Collector v1 : Online
curl http://192.168.99.113:31221/alerts/status
Alerts v1 : Online
curl http://192.168.99.113:31221/sections/1/status
Section 1 v1 : Online
```




DASHBOARD - FIREFOX DEVELOPER EDITION (PRIVATE BROWSING)

192.168.99.113:31221

Dashboard V1

Section 1

timestamp: 2020-02-25T14:35:38.204522Z

gender: male | age: 38-43 | event: exit

Alert

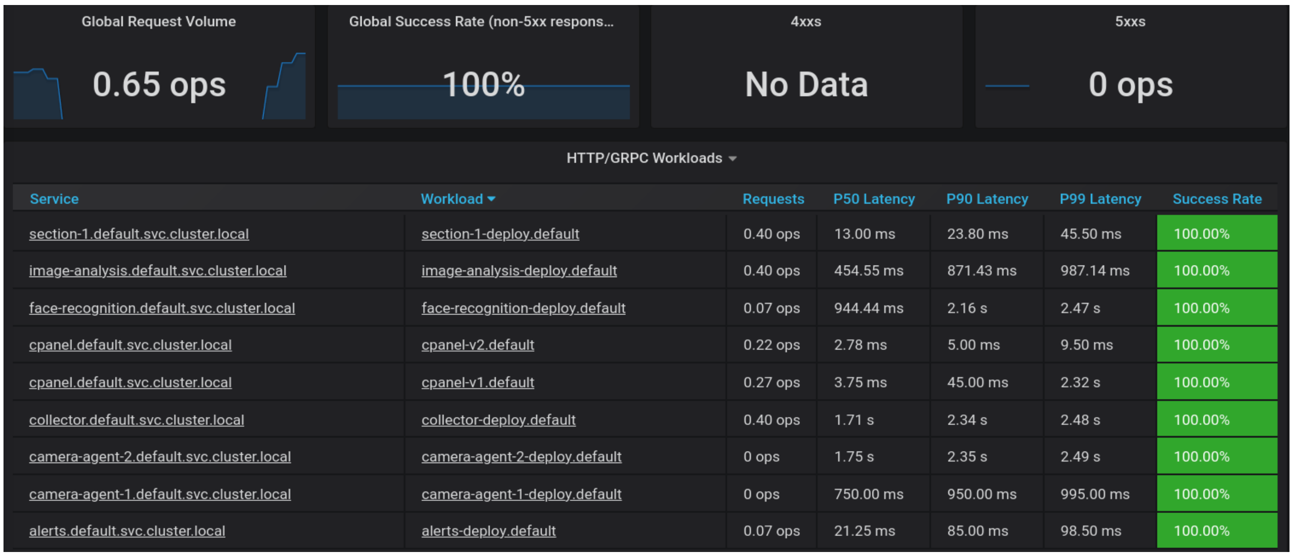
timestamp: 2020-02-25T14:35:27.224857Z

section: 1

event: entry

name: **PersonX**

Default app with cpanel v1 without load:

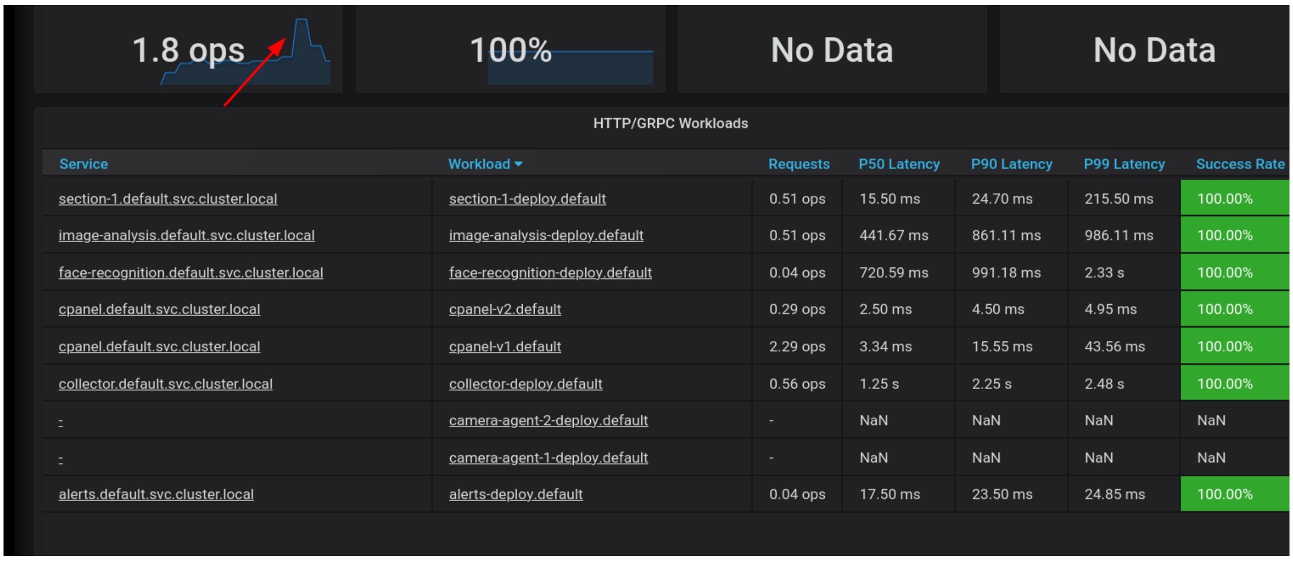


Global Request Volume: 0.65 ops | Global Success Rate (non-5xx respons...: 100% | 4xxs: No Data | 5xxs: 0 ops

HTTP/GRPC Workloads

Service	Workload	Requests	P50 Latency	P90 Latency	P99 Latency	Success Rate
section-1.default.svc.cluster.local	section-1-deploy.default	0.40 ops	13.00 ms	23.80 ms	45.50 ms	100.00%
image-analysis.default.svc.cluster.local	image-analysis-deploy.default	0.40 ops	454.55 ms	871.43 ms	987.14 ms	100.00%
face-recognition.default.svc.cluster.local	face-recognition-deploy.default	0.07 ops	944.44 ms	2.16 s	2.47 s	100.00%
cpanel.default.svc.cluster.local	cpanel-v2.default	0.22 ops	2.78 ms	5.00 ms	9.50 ms	100.00%
cpanel.default.svc.cluster.local	cpanel-v1.default	0.27 ops	3.75 ms	45.00 ms	2.32 s	100.00%
collector.default.svc.cluster.local	collector-deploy.default	0.40 ops	1.71 s	2.34 s	2.48 s	100.00%
camera-agent-2.default.svc.cluster.local	camera-agent-2-deploy.default	0 ops	1.75 s	2.35 s	2.49 s	100.00%
camera-agent-1.default.svc.cluster.local	camera-agent-1-deploy.default	0 ops	750.00 ms	950.00 ms	995.00 ms	100.00%
alerts.default.svc.cluster.local	alerts-deploy.default	0.07 ops	21.25 ms	85.00 ms	98.50 ms	100.00%

```
$ make load
for i in {1..100}; do sleep 0.2; curl http://192.168.99.113:31221/status; printf "\n"; done
CPanel v1 : Online
CPanel v1 : Online
CPanel v1 : Online
```



1.8 ops | 100% | No Data | No Data

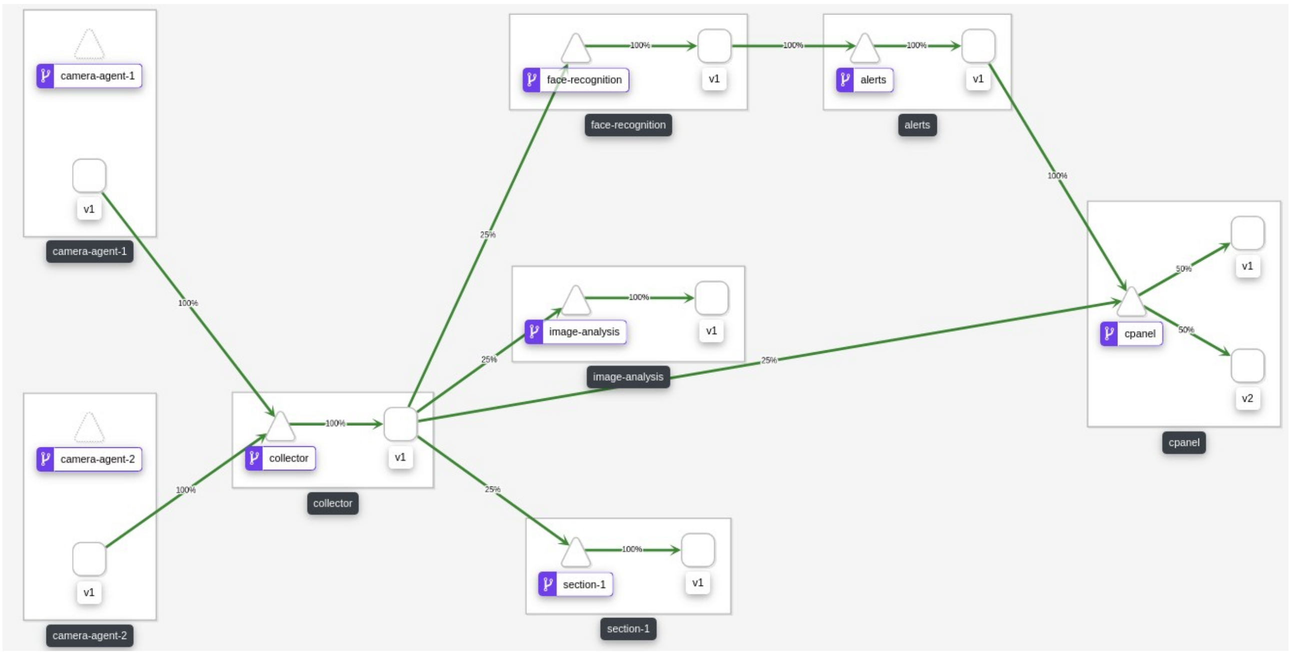
HTTP/GRPC Workloads

Service	Workload	Requests	P50 Latency	P90 Latency	P99 Latency	Success Rate
section-1.default.svc.cluster.local	section-1-deploy.default	0.51 ops	15.50 ms	24.70 ms	215.50 ms	100.00%
image-analysis.default.svc.cluster.local	image-analysis-deploy.default	0.51 ops	441.67 ms	861.11 ms	986.11 ms	100.00%
face-recognition.default.svc.cluster.local	face-recognition-deploy.default	0.04 ops	720.59 ms	991.18 ms	2.33 s	100.00%
cpanel.default.svc.cluster.local	cpanel-v2.default	0.29 ops	2.50 ms	4.50 ms	4.95 ms	100.00%
cpanel.default.svc.cluster.local	cpanel-v1.default	2.29 ops	3.34 ms	15.55 ms	43.56 ms	100.00%
collector.default.svc.cluster.local	collector-deploy.default	0.56 ops	1.25 s	2.25 s	2.48 s	100.00%
-	camera-agent-2-deploy.default	-	NaN	NaN	NaN	NaN
-	camera-agent-1-deploy.default	-	NaN	NaN	NaN	NaN
alerts.default.svc.cluster.local	alerts-deploy.default	0.04 ops	17.50 ms	23.50 ms	24.85 ms	100.00%

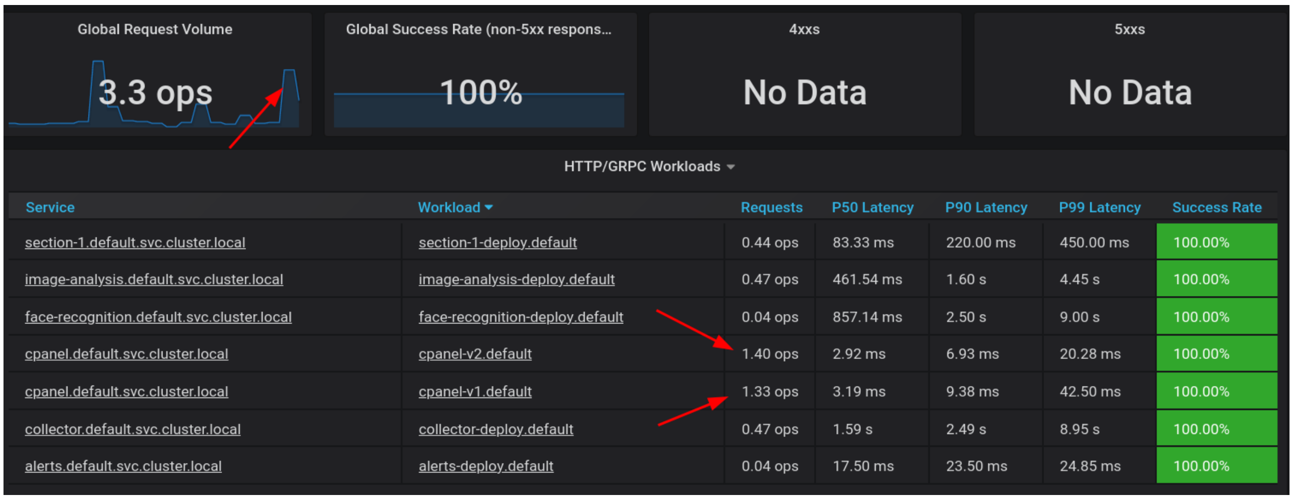
```
$ make cpanel-50-50
./kubectl apply -f istio/virt_svc_50-50.yaml
virtualservice.networking.istio.io/cpanel configured
check configuration
$ k get virtualservices cpanel -o yaml

route:
- destination:
    host: cpanel.default.svc.cluster.local
    port:
      number: 8080
    subset: v1
  weight: 50
- destination:
    host: cpanel.default.svc.cluster.local
    port:
      number: 8080
    subset: v2
  weight: 50
```

```
$ make start-cameras
```



```
$ make load
for i in {1..100}; do sleep 0.2; curl http://192.168.99.113:31221/status; printf "\n"; done
CPanel v1 : Online
CPanel v1 : Online
CPanel v1 : Online
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
```



Global Request Volume	Global Success Rate (non-5xx respons...	4xxs	5xxs
3.3 ops	100%	No Data	No Data

HTTP/GRPC Workloads

Service	Workload	Requests	P50 Latency	P90 Latency	P99 Latency	Success Rate
section-1.default.svc.cluster.local	section-1-deploy.default	0.44 ops	83.33 ms	220.00 ms	450.00 ms	100.00%
image-analysis.default.svc.cluster.local	image-analysis-deploy.default	0.47 ops	461.54 ms	1.60 s	4.45 s	100.00%
face-recognition.default.svc.cluster.local	face-recognition-deploy.default	0.04 ops	857.14 ms	2.50 s	9.00 s	100.00%
cpanel.default.svc.cluster.local	cpanel-v2.default	1.40 ops	2.92 ms	6.93 ms	20.28 ms	100.00%
cpanel.default.svc.cluster.local	cpanel-v1.default	1.33 ops	3.19 ms	9.38 ms	42.50 ms	100.00%
collector.default.svc.cluster.local	collector-deploy.default	0.47 ops	1.59 s	2.49 s	8.95 s	100.00%
alerts.default.svc.cluster.local	alerts-deploy.default	0.04 ops	17.50 ms	23.50 ms	24.85 ms	100.00%

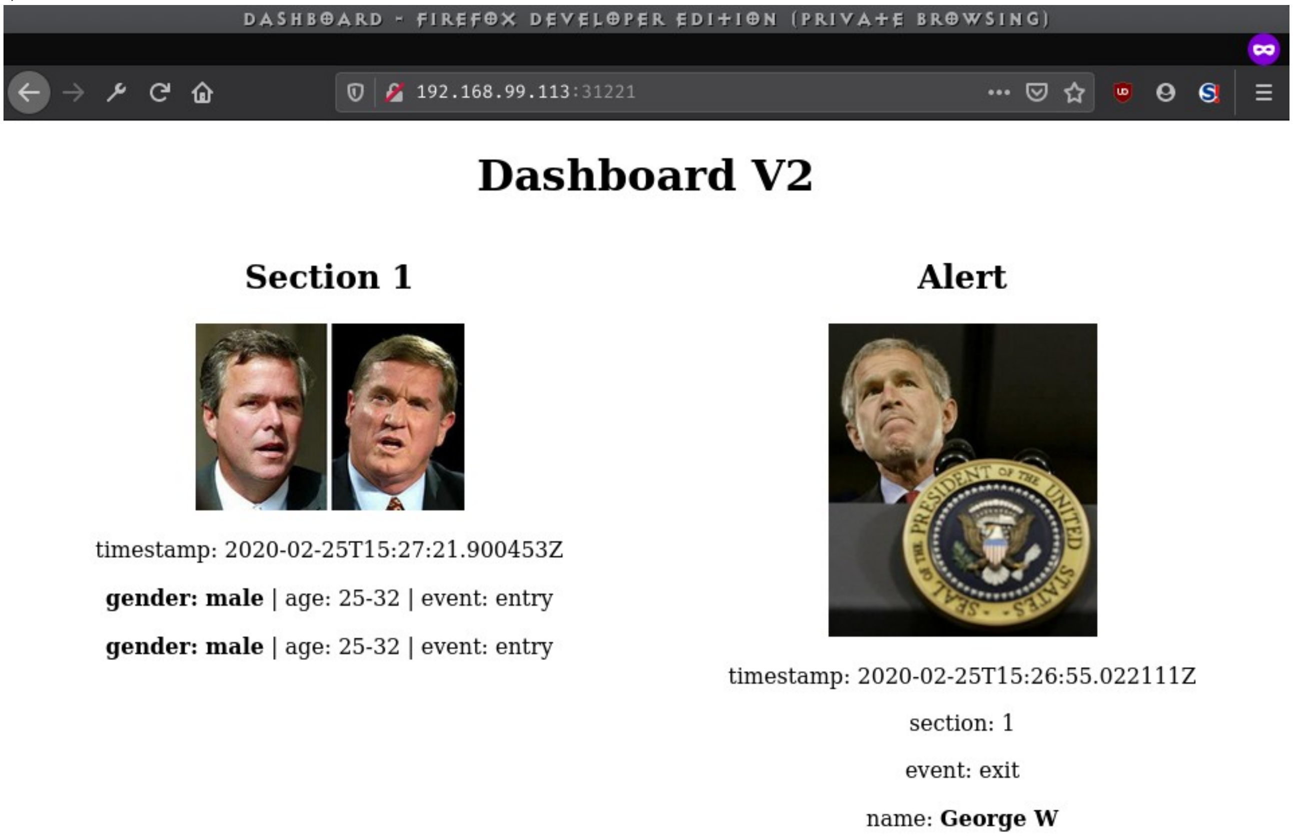
```
$ make cpanel-v2
./kubectl apply -f istio/virt_svc_v2.yaml
virtualservice.networking.istio.io/cpanel configured
```

check configuration

```
$ k get virtualservices cpanel -o yaml

route:
- destination:
    host: cpanel.default.svc.cluster.local
    port:
      number: 8080
    subset: v1
  weight: 0
- destination:
    host: cpanel.default.svc.cluster.local
    port:
      number: 8080
    subset: v2
  weight: 100

$ make start-cameras
```



Dashboard V2

Section 1

timestamp: 2020-02-25T15:27:21.900453Z

gender: male | age: 25-32 | event: entry

gender: male | age: 25-32 | event: entry

Alert

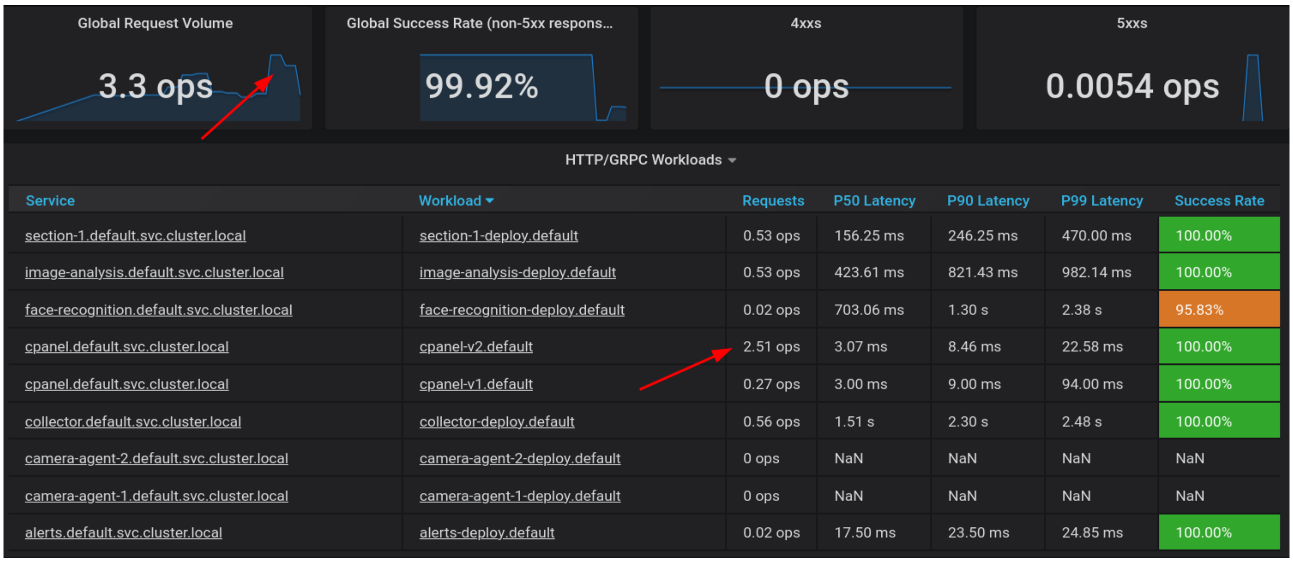
timestamp: 2020-02-25T15:26:55.022111Z

section: 1

event: exit

name: **George W**

```
$ make load
for i in {1..100}; do sleep 0.2; curl http://192.168.99.113:31221/status; printf "\n"; done
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
```



Global Request Volume	Global Success Rate (non-5xx respons...	4xxs	5xxs
3.3 ops	99.92%	0 ops	0.0054 ops

HTTP/GRPC Workloads

Service	Workload	Requests	P50 Latency	P90 Latency	P99 Latency	Success Rate
section-1.default.svc.cluster.local	section-1-deploy.default	0.53 ops	156.25 ms	246.25 ms	470.00 ms	100.00%
image-analysis.default.svc.cluster.local	image-analysis-deploy.default	0.53 ops	423.61 ms	821.43 ms	982.14 ms	100.00%
face-recognition.default.svc.cluster.local	face-recognition-deploy.default	0.02 ops	703.06 ms	1.30 s	2.38 s	95.83%
cpanel.default.svc.cluster.local	cpanel-v2.default	2.51 ops	3.07 ms	8.46 ms	22.58 ms	100.00%
cpanel.default.svc.cluster.local	cpanel-v1.default	0.27 ops	3.00 ms	9.00 ms	94.00 ms	100.00%
collector.default.svc.cluster.local	collector-deploy.default	0.56 ops	1.51 s	2.30 s	2.48 s	100.00%
camera-agent-2.default.svc.cluster.local	camera-agent-2-deploy.default	0 ops	NaN	NaN	NaN	NaN
camera-agent-1.default.svc.cluster.local	camera-agent-1-deploy.default	0 ops	NaN	NaN	NaN	NaN
alerts.default.svc.cluster.local	alerts-deploy.default	0.02 ops	17.50 ms	23.50 ms	24.85 ms	100.00%

Load balancing
Default round robin between v1 and v2 cpanel (should be 1:3)

```
$ make scale_v2_x3
kubectl scale deployment cpanel-v2 --replicas=3
collector-deploy-558dd7dd45-8rlwq         2/2    Running   3       9h
cpanel-v1-8446d9dd45-wx6mz                2/2    Running   2       9h
cpanel-v2-8445ff5964-lgj84                1/2    Running   0       6s
cpanel-v2-8445ff5964-qdhk8                0/2    Running   0       6s
cpanel-v2-8445ff5964-r4r2d                2/2    Running   3       9h
face-recognition-deploy-7b954c454-fdphg   2/2    Running   3       9h
```

Here we can see how kubernetes scales our service.

```
$ make load_balancing
./kubectl apply -f istio/round_robin.yaml

route:
- destination:
    host: cpanel.default.svc.cluster.local
    port:
      number: 8080

$ make load
for i in {1..100}; do sleep 0.2; curl http://192.168.99.113:31221/status; printf "\n"; done
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
CPanel v2 : Online
CPanel v1 : Online
CPanel v2 : Online
CPanel v1 : Online

$ make random
```

```
./kubectl apply -f istio/random_lb.yaml
destinationrule.networking.istio.io/cpanel configured
$ k get destinationrules cpanel -o yaml

$ make load
for i in {1..100}; do sleep 0.2; curl http://192.168.99.113:31221/status; printf "\n"; done
CPanel v2 : Online
CPanel v2 : Online
CPanel v1 : Online
CPanel v2 : Online
CPanel v1 : Online
CPanel v2 : Online
CPanel v1 : Online
CPanel v2 : Online

$ make all-reset
./kubectl delete service --all
```