Event-driven architecture

Unpacking Knative

Jeff Barnes
GC Micro-mission

Event-driven (Serverless) Architecture

Event-Driven, sometimes also called serverless or functions as a service, is a computing execution model in which the infrastructure/provider dynamically manages the allocation of machine resources

- Code is deployed to a CSP or elsewhere
- Instance does not exist until invoked, spins up and scales as required, scales to zero when complete
- Changes IT requirements;
    - Security
    - Administration (patching, deployment)
    - Architecture
    - Cold start
- Currently various standards and little interoperability between CSP's
- Specific use cases include 'intents' for voice activated assistants (AWS Echo skills deployed in AWS Lambda)

IBM OpenWhisk

AWS Lambda

Google Cloud Functions

Azure Functions

.....

Event-driven (Serverless) Architecture

Originally found in public clouds, CSP instances are heading on-prem via appliance (Google Cloud Services Platform, Azure Stack, AWS Greengrass). A number of open-source versions are now available which can be deployed on-prem or in public cloud.

Examples include knative, OpenFaaS and Kubeless. This deck unpacks Knative, which can be used to auto-scale serverless-style functions, applications, and containers on Kubernetes

Knative OpenFaaS Kubeless

Initially serverless, Functions as a service (FaaS) and event-driven architecture referred to a microservice that is run 'on public cloud' compute engine only when invoked. Recently, serverless is also being used to describe managed container service (CaaS) and application deployments where the user is not responsible for the compute where/when the container runs (Azure AKS and ACI, AWS EKS and Fargate, Google GKE, Google App Engine, Ibm Kubernetes Service....)

Knative

Knative can be used to deploy serverless-style functions, applications, and containers to an auto-scaling runtime on Kubernetes. Also has many other features: deploy multiple versions, perform custom tasks on your application's source code, build reusable templates…

- Install Knative on an Istio Cluster (Istio injection enabled)
- Select and install demo app (helloworld-go)

Invoke the app

- Determine if running ‘LoadBalancer’ or ‘NodePort’

```
export IP_ADDRESS=$(kubectl get svc istio-ingressgateway --namespace istio-system --output
'jsonpath={.status.loadBalancer.ingress[0].ip}')
```

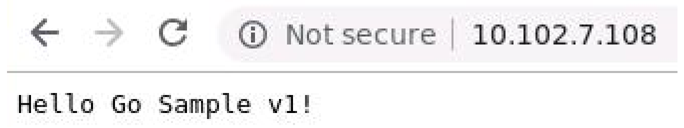
OR

```
export IP_ADDRESS=$(kubectl get node  --output
'jsonpath={.items[0].status.addresses[0].address}'):$(kubectl get svc istio-ingressgateway --
namespace istio-system   --output 'jsonpath={.spec.ports[?(@.port==80)].nodePort}')

export HOST_URL=$(kubectl get ksvc helloworld-go  --output jsonpath='{.status.domain}')

curl -H "Host: ${HOST_URL}" http://${IP_ADDRESS}
Hello World: Go Sample v1!
```

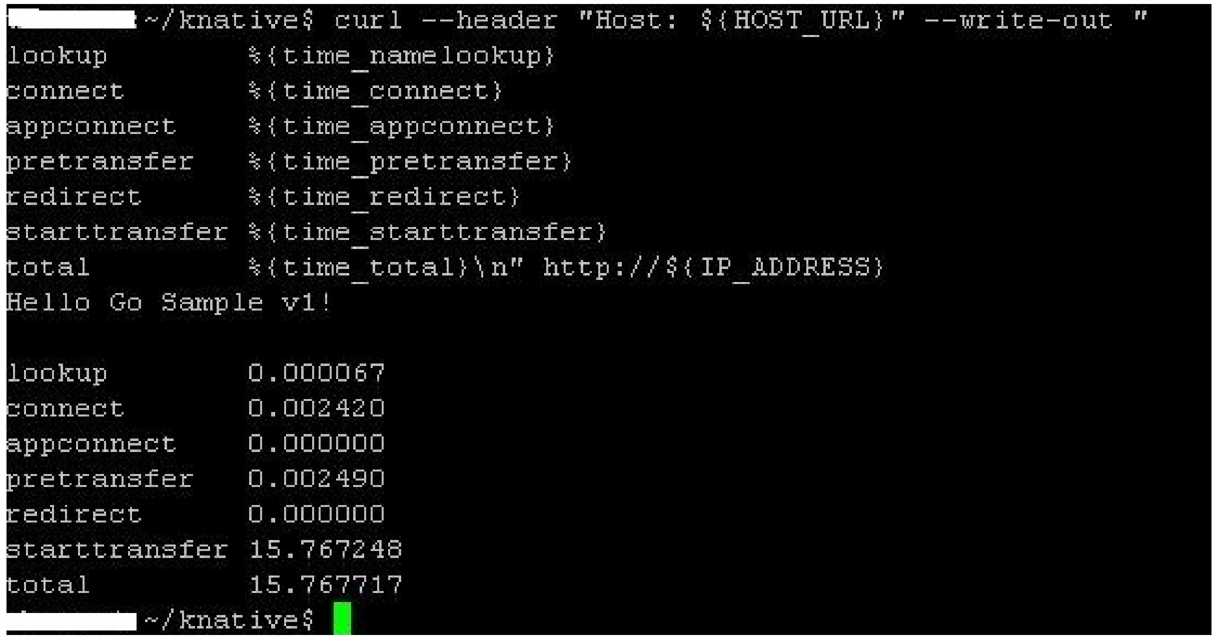
Knative – Cold Start

```
~/knative$ curl --header "Host: ${HOST_URL}" --write-out "
> lookup        %{time_namelookup}
> connect       %{time_connect}
> appconnect    %{time_appconnect}
> pretransfer   %{time_pretransfer}
> redirect      %{time_redirect}
> starttransfer %{time_starttransfer}
> total         %{time_total}\n" \
> http://${IP_ADDRESS}
Hello Go Sample v1!

lookup        0.000067
connect       0.002420
appconnect    0.000000
pretransfer   0.002490
redirect      0.000000
starttransfer 15.767278
total         15.767717
```

Cold Start time 15 sec (need to fix it in my demo)



```
Hello Go Sample v1!
```

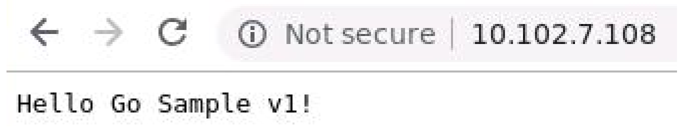


```
[redacted] ~/knative$ curl --header "Host: ${HOST_URL}" --write-out "
lookup        %{time_namelookup}
connect       %{time_connect}
appconnect    %{time_appconnect}
pretransfer   %{time_pretransfer}
redirect      %{time_redirect}
starttransfer %{time_starttransfer}
total         %{time_total}\n" http://${IP_ADDRESS}
Hello Go Sample v1!

lookup        0.000067
connect       0.002420
appconnect    0.000000
pretransfer   0.002490
redirect      0.000000
starttransfer 15.767248
total         15.767717
[redacted] ~/knative$
```

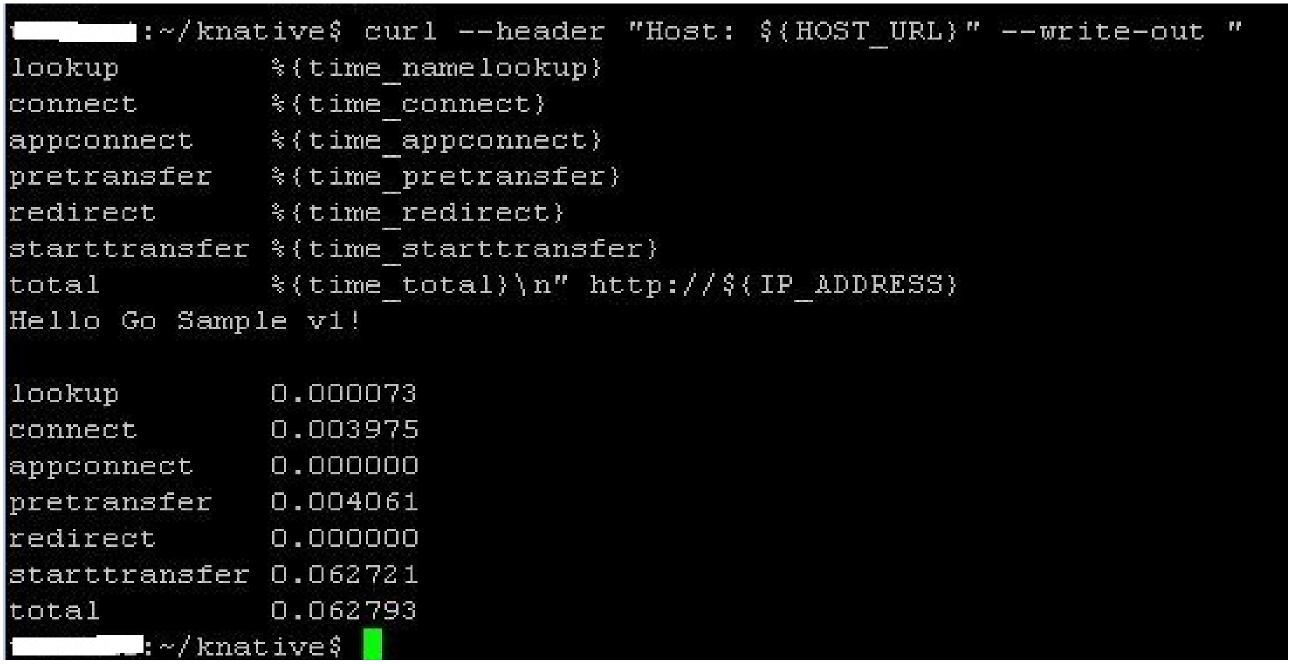
Knative – Subsequent Start

```
~/knative$ curl --header "Host: ${HOST_URL}" --write-out "
lookup        %{time_namelookup}
connect       %{time_connect}
appconnect    %{time_appconnect}
pretransfer   %{time_pretransfer}
redirect      %{time_redirect}
starttransfer %{time_starttransfer}
total         %{time_total}\n" http://${IP_ADDRESS}
Hello Go Sample v1!

lookup        0.000073
connect       0.003975
appconnect    0.000000
pretransfer   0.004061
redirect      0.000000
starttransfer 0.062721
total         0.062793
```

Subsequent response time < 1s

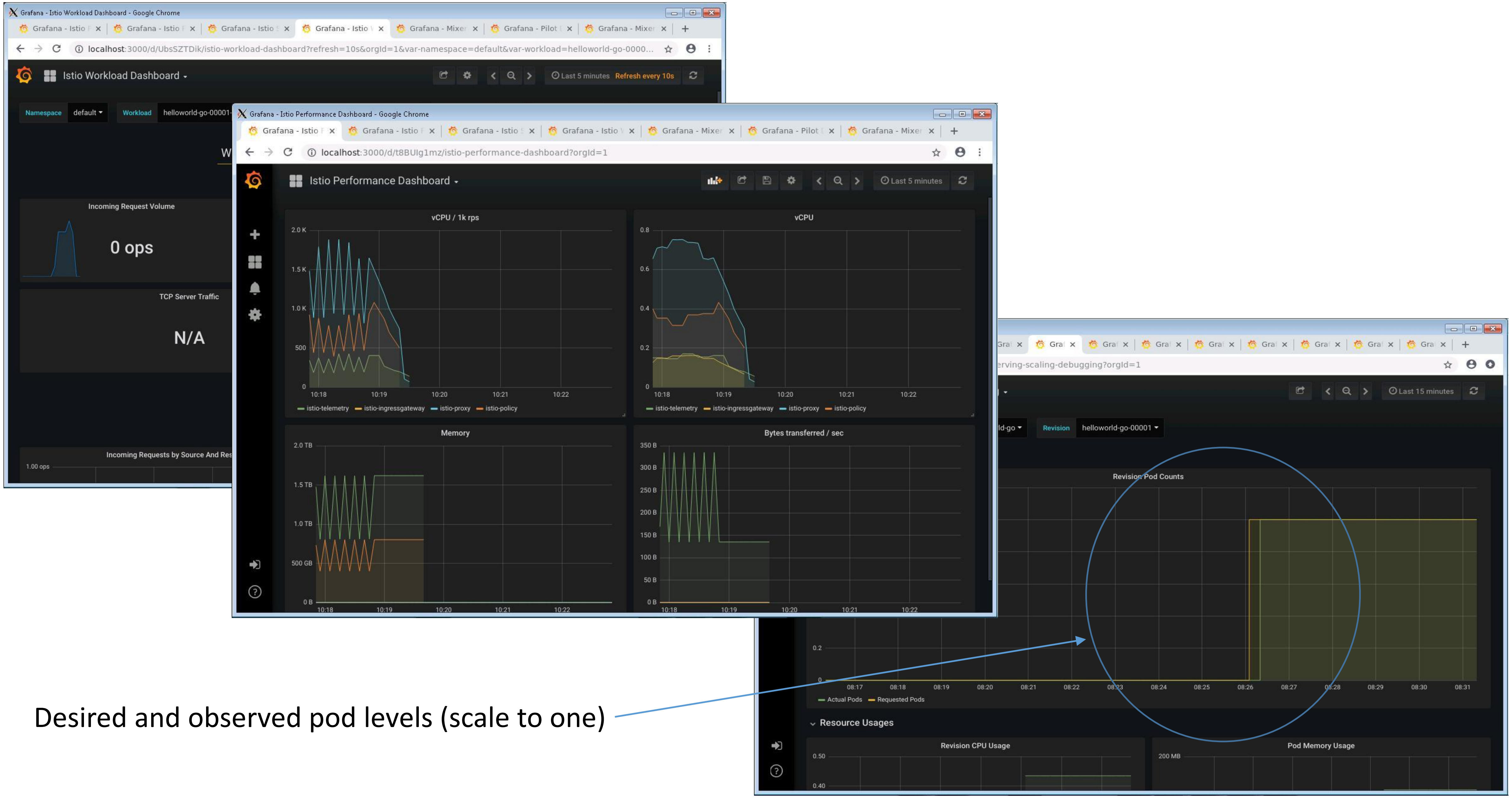


```
Hello Go Sample v1!
```



```
[redacted]:~/knative$ curl --header "Host: ${HOST_URL}" --write-out "
lookup        %{time_namelookup}
connect       %{time_connect}
appconnect    %{time_appconnect}
pretransfer   %{time_pretransfer}
redirect      %{time_redirect}
starttransfer %{time_starttransfer}
total         %{time_total}\n" http://${IP_ADDRESS}
Hello Go Sample v1!

lookup        0.000073
connect       0.003975
appconnect    0.000000
pretransfer   0.004061
redirect      0.000000
starttransfer 0.062721
total         0.062793
[redacted]:~/knative$
```

Knative – Performance Monitoring Grafana

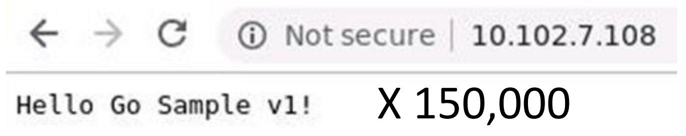

Grafana - Istio Workload Dashboard - Google Chrome
localhost:3000/d/UbsSZTDik/istio-workload-dashboard?refresh=10s&orgId=1&var-namespace=default&var-workload=helloworld-go-0000...
Istio Workload Dashboard
Last 5 minutes Refresh every 10s
Namespace default
Workload helloworld-go-00001
Incoming Request Volume
0 ops
TCP Server Traffic
N/A
Incoming Requests by Source And Res
Grafana - Istio Performance Dashboard - Google Chrome
localhost:3000/d/t8BUIg1mz/istio-performance-dashboard?orgId=1
Istio Performance Dashboard
Last 5 minutes
vCPU / 1k rps
vCPU
Memory
Bytes transferred / sec
istio-telemetry
istio-ingressgateway
istio-proxy
istio-policy
erving-scaling-debugging?orgId=1
Last 15 minutes
Revision helloworld-go-00001
Revision Pod Counts
Actual Pods
Requested Pods
Resource Usages
Revision CPU Usage
Pod Memory Usage

Desired and observed pod levels (scale to one)

Knative – Scale based on workload

hey runs number of requests in the provided concurrency level and prints stats.

-n Number of requests to run. Default is 200.
-c Number of requests to run concurrently. Total number of requests cannot be smaller than the concurrency level. Default is 50.



Not secure | 10.102.7.108

```
Hello Go Sample v1!
```

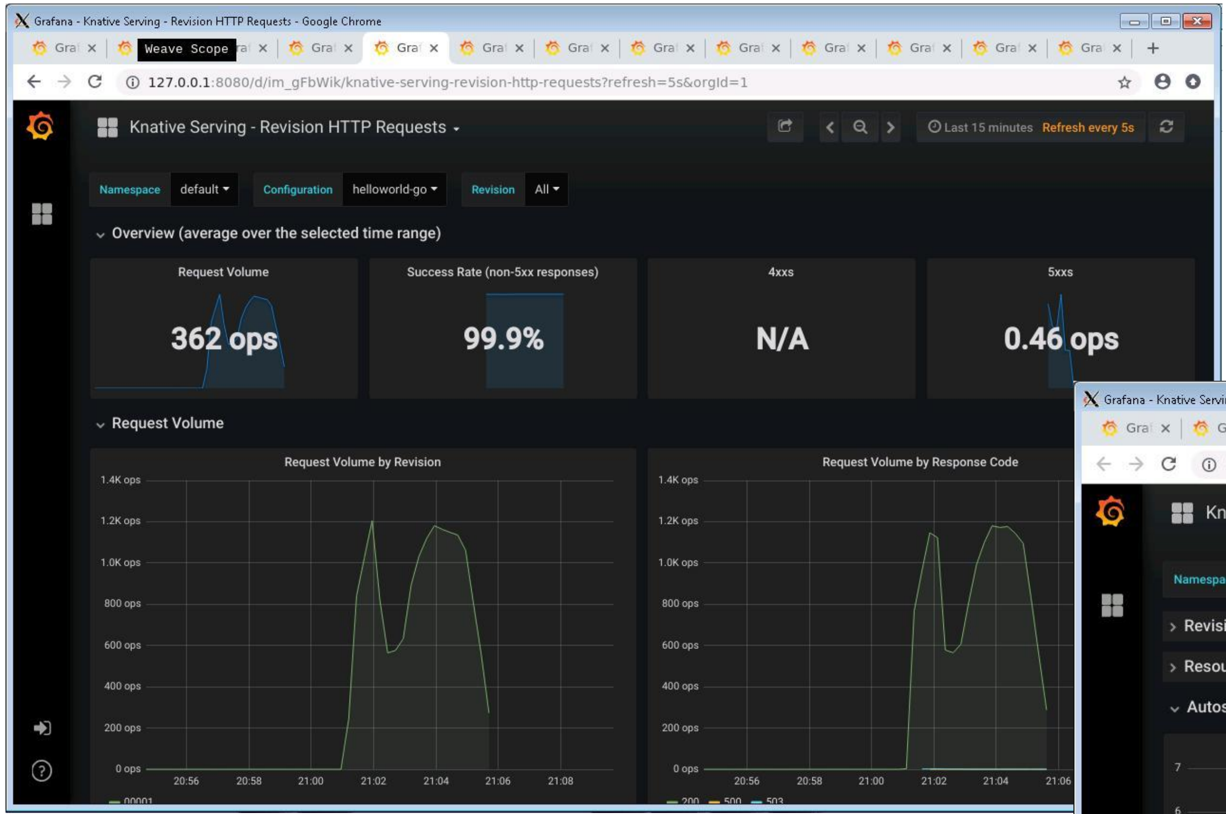
X 150,000

Send 150,000 requests (with 500 requests in parallel), each taking 1 second

```
~/go/bin$ ./hey -host helloworld-go.default.example.com -c 500 -n
150000   "http://${IP_ADDRESS?}?sleep=1000"

Summary:
  Total:        291.8676 secs
  Slowest:      17.4429 secs
  Fastest:      0.0025 secs
  Average:      0.9101 secs
  Requests/sec: 513.9316

  Total data:   3002220 bytes
  Size/request: 20 bytes

Response time histogram:
  0.002 [1]      |
  1.747 [116370]        |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  3.491 [19319] |■■■■■■■
  5.235 [9837]  |■■■
  6.979 [3112]  |■
  8.723 [971]   |
  10.467 [208]  |
  12.211 [122]  |
  13.955 [23]   |
  15.699 [24]   |
  17.443 [13]   |
```

```
Latency distribution:
  10% in 0.0041 secs
  25% in 0.0050 secs
  50% in 0.0083 secs
  75% in 1.3462 secs
  90% in 3.4178 secs
  95% in 4.5164 secs
  99% in 6.8141 secs

Details (average, fastest, slowest):
  DNS+dialup:   0.0002 secs, 0.0025 secs, 17.4429 secs
  DNS-lookup:   0.0000 secs, 0.0000 secs, 0.0000 secs
  req write:    0.0002 secs, 0.0000 secs, 0.1051 secs
  resp wait:    0.9091 secs, 0.0024 secs, 17.4428 secs
  resp read:    0.0005 secs, 0.0000 secs, 0.1377 secs

Status code distribution:
  [200] 149940 responses
  [503] 60 responses
```

Knative – Scale based on workload w/Performance Monitoring Grafana



Grafana - Knative Serving - Revision HTTP Requests - Google Chrome

127.0.0.1:8080/d/im_gFbWik/knative-serving-revision-http-requests?refresh=5s&orgId=1

Knative Serving - Revision HTTP Requests

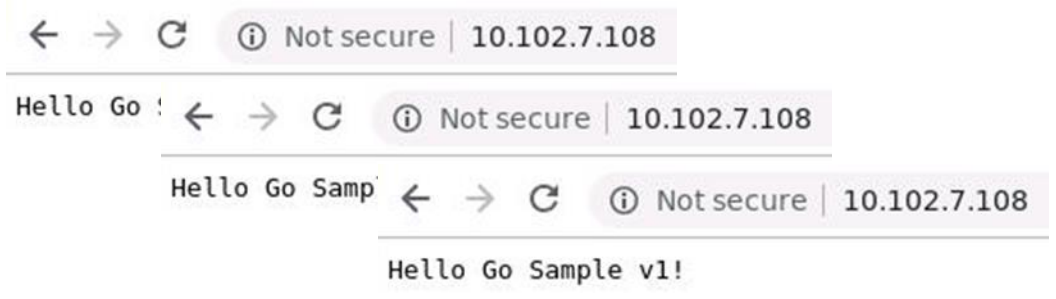
Last 15 minutes Refresh every 5s

Namespace: default | Configuration: helloworld-go | Revision: All

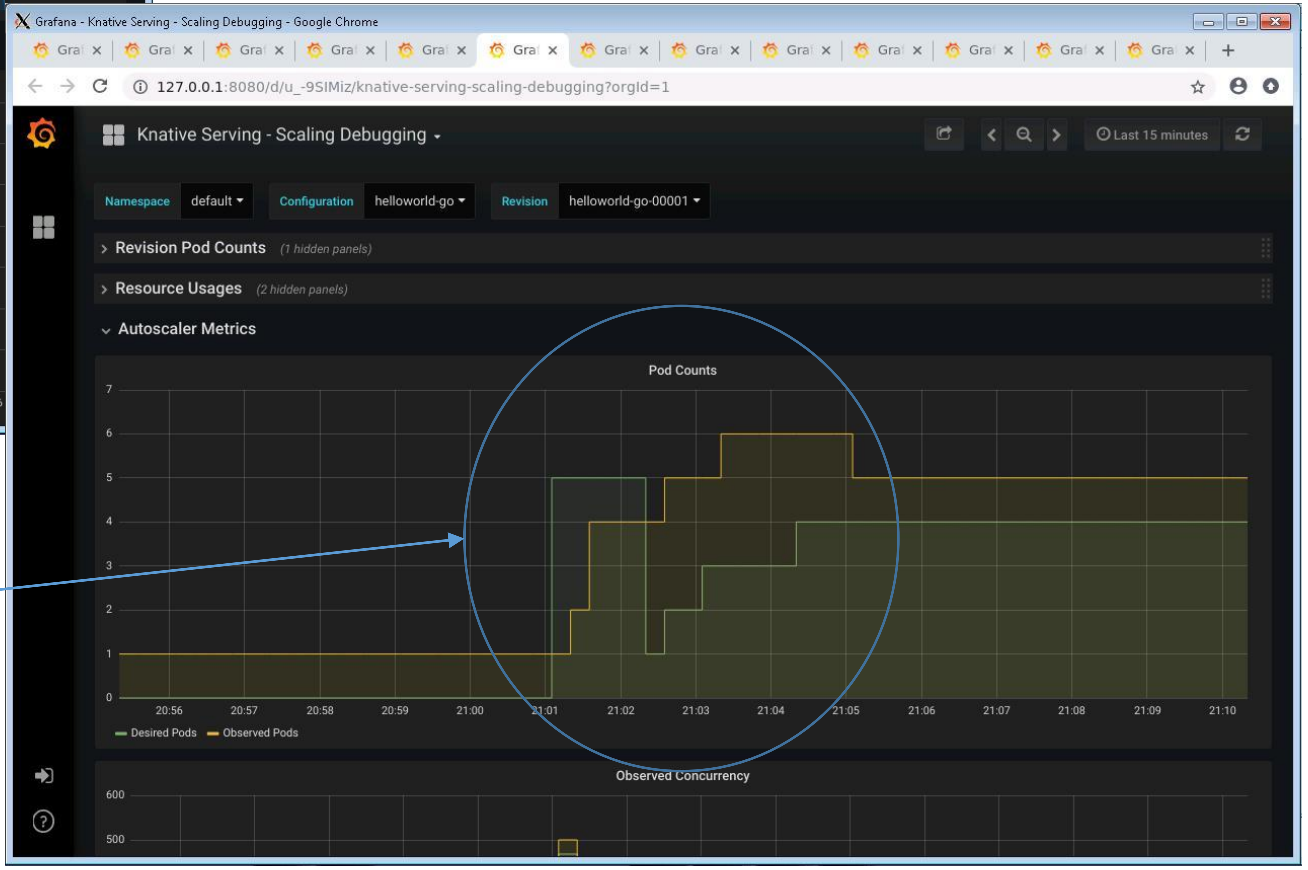
Overview (average over the selected time range)

Request Volume	Success Rate (non-5xx responses)	4xxs	5xxs
362 ops	99.9%	N/A	0.46 ops

Request Volume



```
Not secure | 10.102.7.108
Hello Go S
Not secure | 10.102.7.108
Hello Go Samp
Not secure | 10.102.7.108
Hello Go Sample v1!
```



Grafana - Knative Serving - Scaling Debugging - Google Chrome

127.0.0.1:8080/d/u_-9SIMiz/knative-serving-scaling-debugging?orgId=1

Knative Serving - Scaling Debugging

Last 15 minutes

Namespace: default | Configuration: helloworld-go | Revision: helloworld-go-00001

Revision Pod Counts (1 hidden panels)

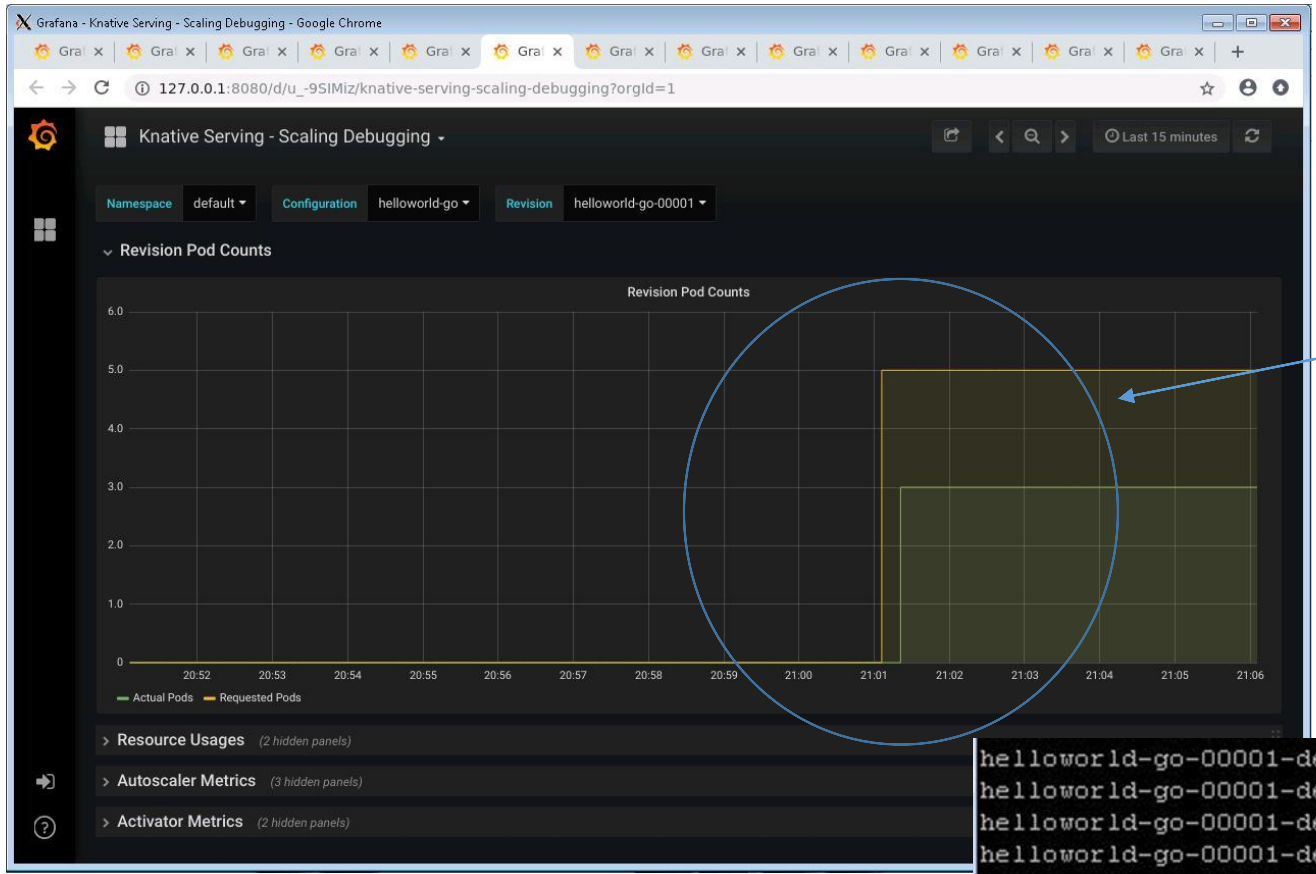
Resource Usages (2 hidden panels)

Autoscaler Metrics

Desired and observed pod levels

Knative – Scale based on workload

Knative Serving, by default, has a concurrent requests target of 100. Sending 500 concurrent requests causes autoscaling to note that it needs to run 5 Pods to satisfy this level



Grafana - Knative Serving - Scaling Debugging - Google Chrome

127.0.0.1:8080/d/u_-9SIMiz/knative-serving-scaling-debugging?orgId=1

Knative Serving - Scaling Debugging

Namespace: default · Configuration: helloworld-go · Revision: helloworld-go-00001

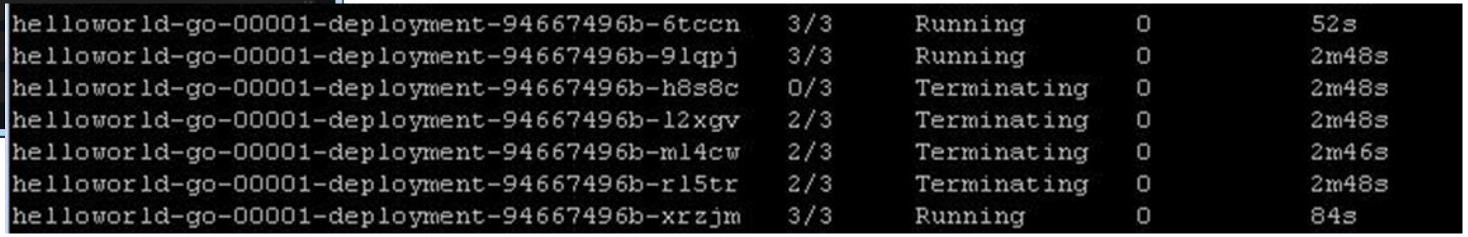
Revision Pod Counts

Resource Usages (2 hidden panels)

Autoscaler Metrics (3 hidden panels)

Activator Metrics (2 hidden panels)

Desired and observed pod levels



```
helloworld-go-00001-deployment-94667496b-6tccn   3/3   Running       0   52s
helloworld-go-00001-deployment-94667496b-9lqpj   3/3   Running       0   2m48s
helloworld-go-00001-deployment-94667496b-h8s8c   0/3   Terminating   0   2m48s
helloworld-go-00001-deployment-94667496b-l2xgv   2/3   Terminating   0   2m48s
helloworld-go-00001-deployment-94667496b-ml4cw   2/3   Terminating   0   2m46s
helloworld-go-00001-deployment-94667496b-rl5tr   2/3   Terminating   0   2m48s
helloworld-go-00001-deployment-94667496b-xrzjm   3/3   Running       0   84s
```

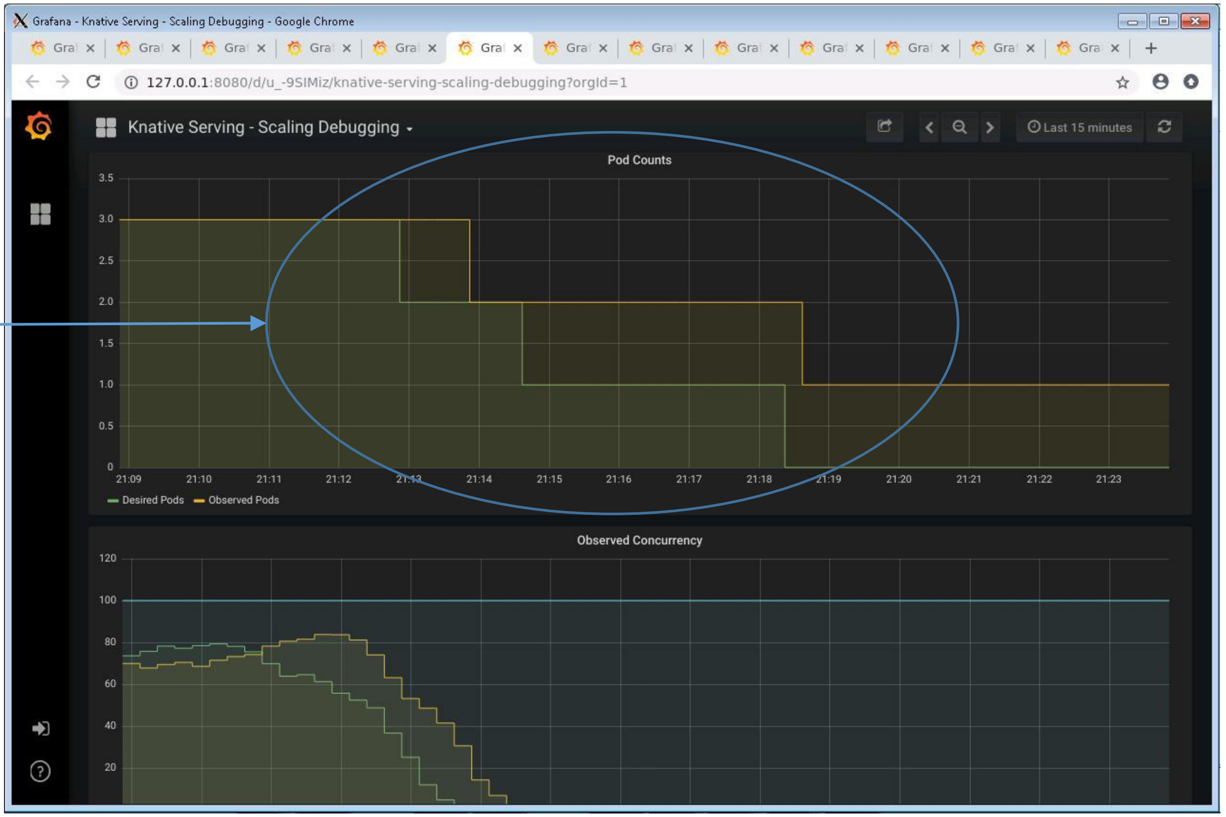
Running and terminating Pods
(as required)

Knative – Scale based on workload w/Performance Monitoring Grafana

```
helloworld-go-00001-deployment-94667496b-2npf2   0/3     Terminating    0          7m20s
helloworld-go-00001-deployment-94667496b-6tccn   3/3     Running        0          8m46s
helloworld-go-00001-deployment-94667496b-xrzjm   3/3     Running        0          9m6s
```

Running and terminating pods as traffic decreases

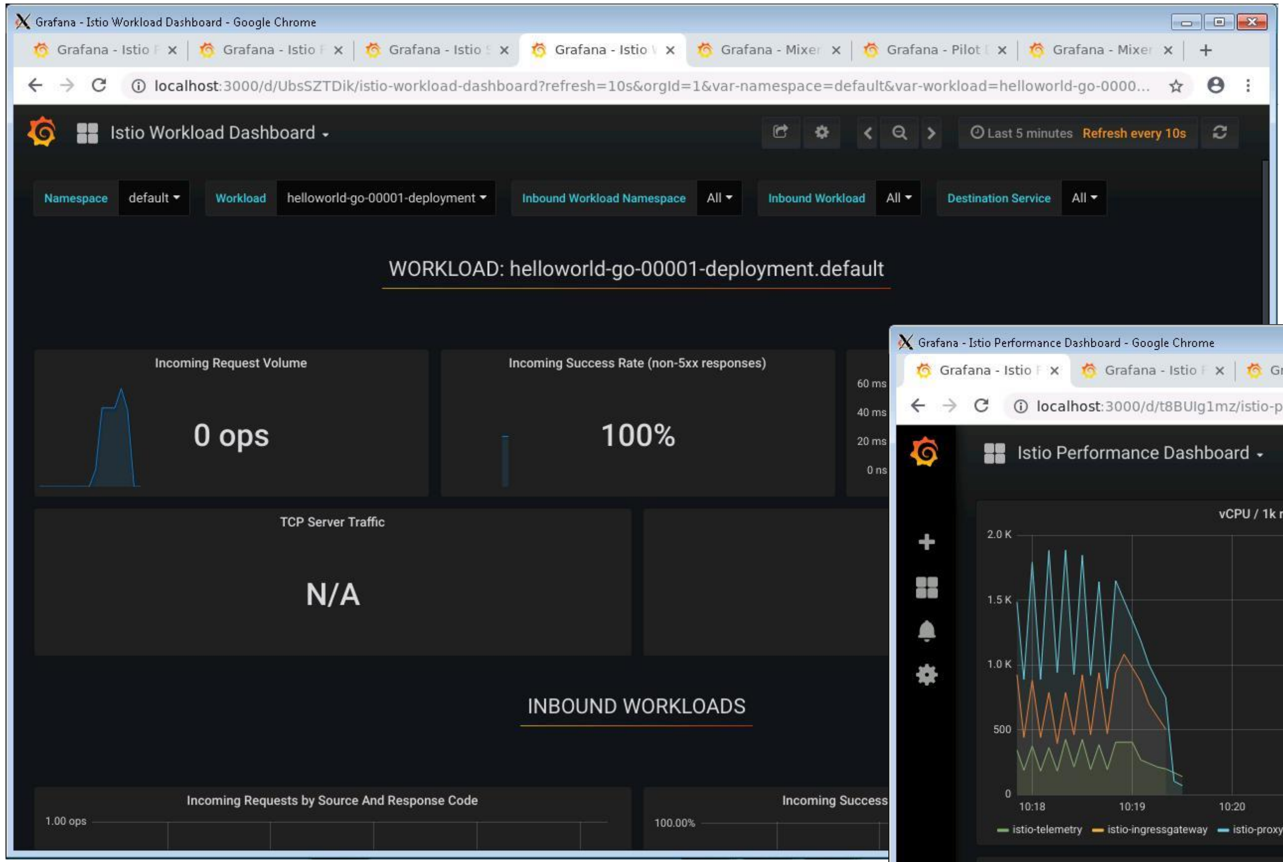
Desired and observed pod levels as traffic decreases


Grafana - Knative Serving - Scaling Debugging - Google Chrome
127.0.0.1:8080/d/u_-9SIMiz/knative-serving-scaling-debugging?orgId=1
Knative Serving - Scaling Debugging
Last 15 minutes
Pod Counts
Desired Pods
Observed Pods
Observed Concurrency

Desired and observed concurrency

```
helloworld-go-00001-deployment-94667496b-xrzjm   2/3     Terminating    0          18m
```

Knative – Performance Monitoring Grafana



Grafana - Istio Workload Dashboard - Google Chrome

localhost:3000/d/UbsSZTDik/istio-workload-dashboard?refresh=10s&orgId=1&var-namespace=default&var-workload=helloworld-go-0000...

Istio Workload Dashboard

Last 5 minutes Refresh every 10s

Namespace: default
Workload: helloworld-go-00001-deployment
Inbound Workload Namespace: All
Inbound Workload: All
Destination Service: All

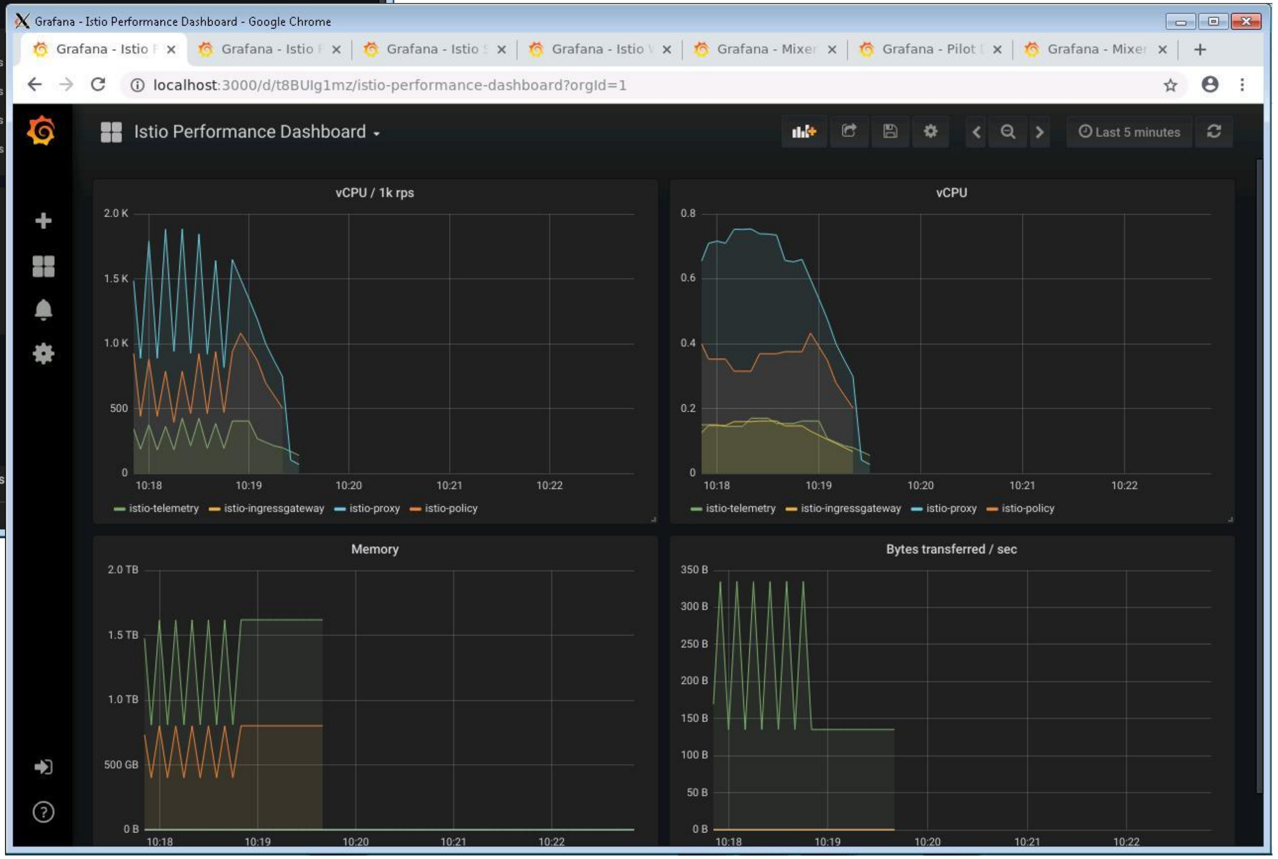
WORKLOAD: helloworld-go-00001-deployment.default

Incoming Request Volume: 0 ops

Incoming Success Rate (non-5xx responses): 100%

TCP Server Traffic: N/A

INBOUND WORKLOADS



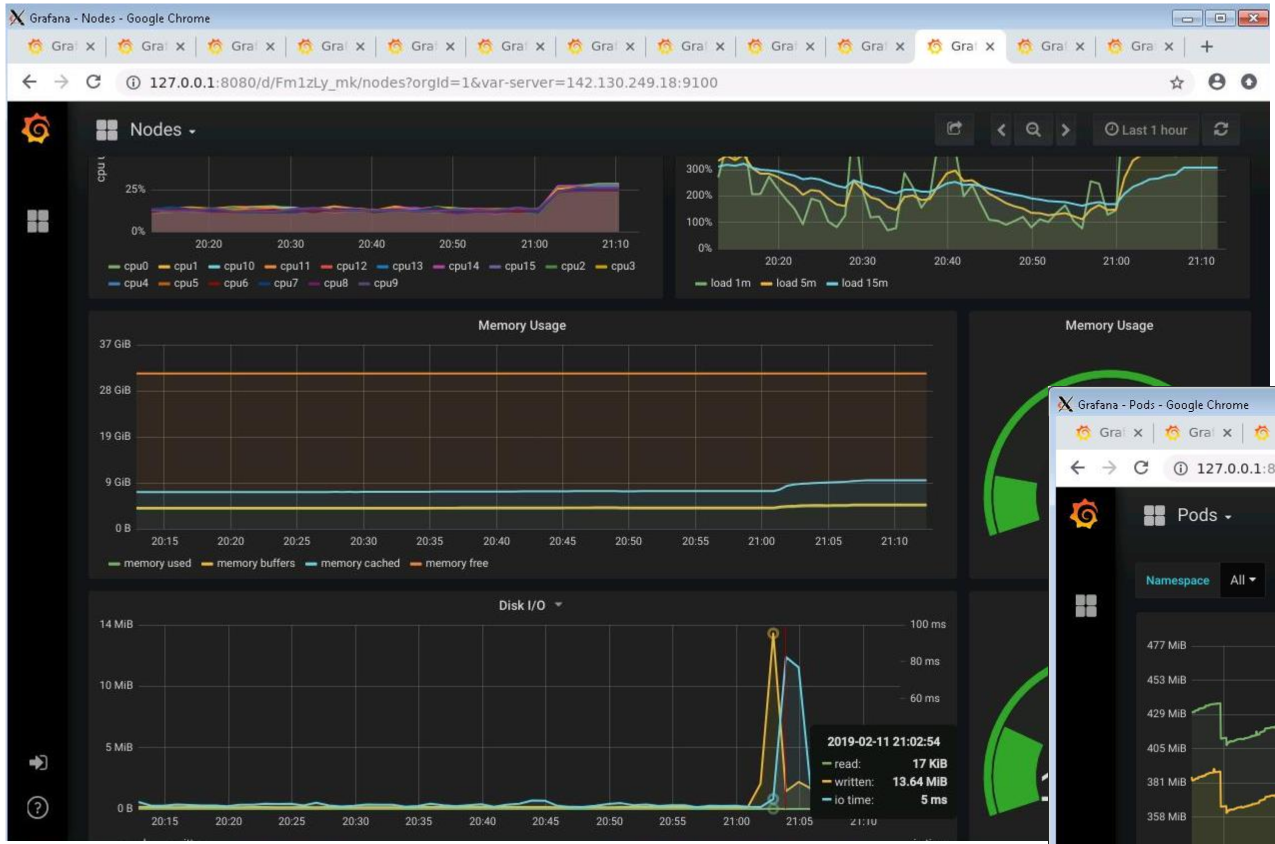
Grafana - Istio Performance Dashboard - Google Chrome

localhost:3000/d/t8BUIg1mz/istio-performance-dashboard?orgId=1

Istio Performance Dashboard

Last 5 minutes

Knative – Performance Monitoring Grafana


Grafana - Nodes - Google Chrome

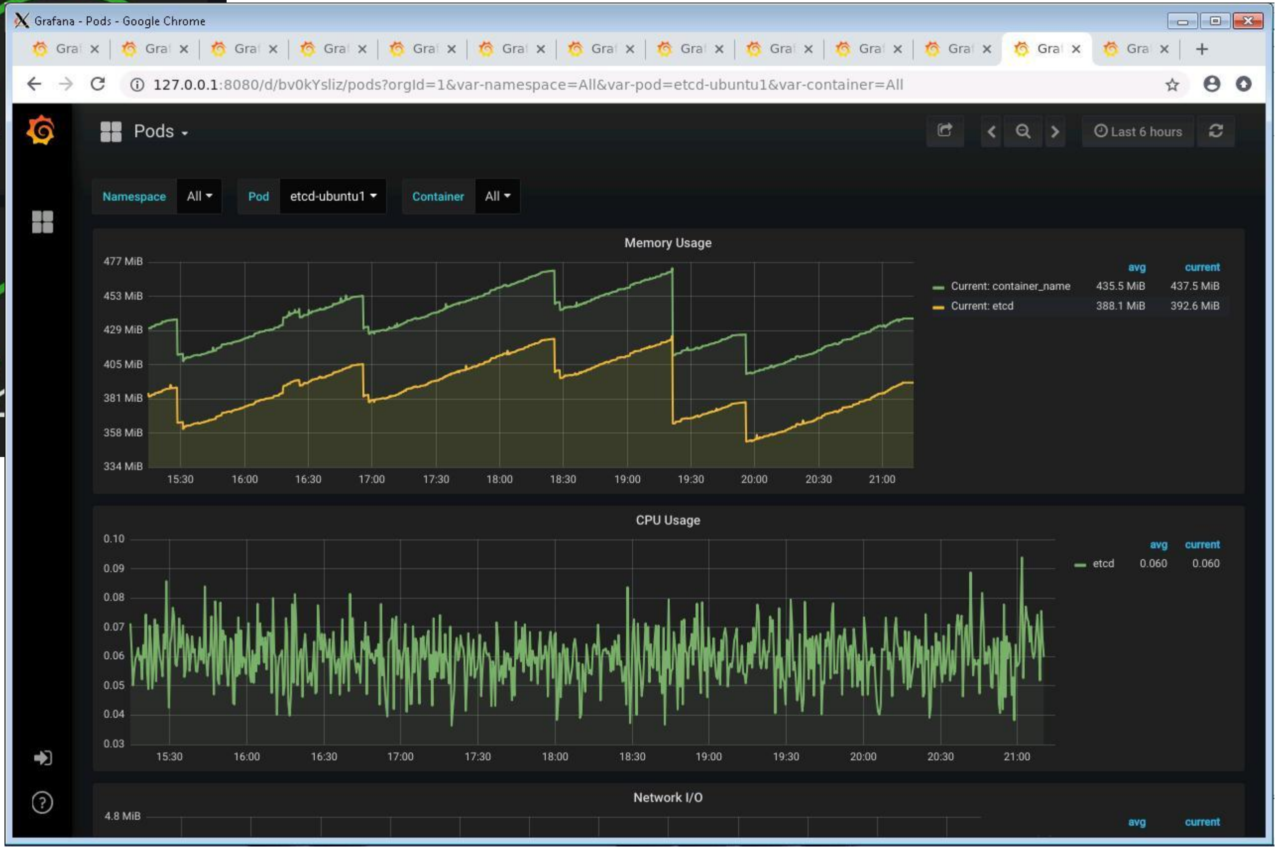
127.0.0.1:8080/d/Fm1zLy_mk/nodes?orgId=1&var-server=142.130.249.18:9100

Nodes

Last 1 hour

2019-02-11 21:02:54
read: 17 KiB
written: 13.64 MiB
io time: 5 ms

cpu0 cpu1 cpu10 cpu11 cpu12 cpu13 cpu14 cpu15 cpu2 cpu3 cpu4 cpu5 cpu6 cpu7 cpu8 cpu9
load 1m load 5m load 15m
Memory Usage
memory used memory buffers memory cached memory free
Disk I/O


Grafana - Pods - Google Chrome

127.0.0.1:8080/d/bv0kYsliz/pods?orgId=1&var-namespace=All&var-pod=etcd-ubuntu1&var-container=All

Pods

Last 6 hours

Namespace: All
Pod: etcd-ubuntu1
Container: All

Memory Usage

	avg	current
Current: container_name	435.5 MiB	437.5 MiB
Current: etcd	388.1 MiB	392.6 MiB

CPU Usage

	avg	current
etcd	0.060	0.060

Knative – Demo

$ kubectl get pods

Runs as required

```
helloworld-go-00001-deployment-94667496b-68x2q   0/3     Init:0/1          0          7s
helloworld-go-00001-deployment-94667496b-9p2cq   0/3     Init:0/1          0          5s
helloworld-go-00001-deployment-94667496b-9sff9   0/3     PodInitializing   0          7s
helloworld-go-00001-deployment-94667496b-jqbjp   0/3     PodInitializing   0          7s
helloworld-go-00001-deployment-94667496b-pb2zq   0/3     Init:0/1          0          7s

helloworld-go-00001-deployment-94667496b-68x2q   0/3     Init:0/1          0          11s
helloworld-go-00001-deployment-94667496b-9p2cq   0/3     PodInitializing   0          9s
helloworld-go-00001-deployment-94667496b-9sff9   2/3     Running           0          11s
helloworld-go-00001-deployment-94667496b-jqbjp   0/3     PodInitializing   0          11s
helloworld-go-00001-deployment-94667496b-pb2zq   0/3     Init:0/1          0          11s
```

Then self-terminates when no longer required

```
helloworld-go-00001-deployment-94667496b-68x2q   2/3     Terminating   0          95s
helloworld-go-00001-deployment-94667496b-9p2cq   2/3     Terminating   0          93s
helloworld-go-00001-deployment-94667496b-9sff9   2/3     Terminating   0          95s
helloworld-go-00001-deployment-94667496b-jqbjp   3/3     Running       0          95s
helloworld-go-00001-deployment-94667496b-pb2zq   2/3     Terminating   0          95s

helloworld-go-00001-deployment-94667496b-68x2q   2/3     Terminating   0          2m47s
helloworld-go-00001-deployment-94667496b-9p2cq   2/3     Terminating   0          2m45s
helloworld-go-00001-deployment-94667496b-9sff9   2/3     Terminating   0          2m47s
helloworld-go-00001-deployment-94667496b-jqbjp   2/3     Terminating   0          2m47s
helloworld-go-00001-deployment-94667496b-pb2zq   2/3     Terminating   0          2m47s
```

Knative – Demo