Comprehensive Container-Based Service Monitoring with Kubernetes and Istio

SREcon Asia Australia 2018-06-06
Fred Moyer
Monitoring Nerd
@phredmoyer

Developer Evangelist
@Circonus / @IRONdb

@IstioMesh Geek

Observability and Statistics Dork
Talk Agenda

- Istio Overview
- Service Level Objectives
- RED Dashboard
- Histogram Telemetry
- Istio Metrics Adapter
- Asking the Right Questions
Istio.io

“An open platform to connect, manage, and secure microservices”
Happy Birthday!
K8S + Istio

- Orchestration
- Deployment
- Scaling
- Data Plane

- Policy Enforcement
- Traffic Management
- Telemetry
- Control Plane
Istio GCP Deployment

Deployment name
istio-test-cluster

installIstioRelease
0.6.0

Baseline GKE Cluster config
GKE Cluster Name
istio-cluster

Zone
us-central1-a

Number of GKE nodes to run
4

Node Machine Type
2 vCPUs 7.5 GB memory

Monitoring, Logging and Tracing
Metrics, Logs, and Traces
Add optional Metrics, Logs, and Traces related plugins to the cluster
- Enable Prometheus for metrics/logs collection
- Enable Grafana for metrics display
- Enable Zipkin for tracing
- Enable ServiceGraph for deployment visualization

Security
Security
Add optional Security related plugins to the cluster
- Enable Automatic Istio sidecar injection
- Enable mutualTLS authentication

Install Applications
- Add BookInfo Sample Application

Deploy
Istio Sample App

$ istioctl create -f apps/bookinfo.yaml
The Comedy of Errors

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

Paperback:
200 pages
Publisher:
Publisher A
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567980

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!
— Reviewer1 Affiliation1

Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.
— Reviewer2 Affiliation2

precedence: 1
route:
- tags:
  - version: v1
  - weight: 100
Go refresh the page
kind: Deployment
metadata:
  name: ratings-v1
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: ratings
        version: v1
    spec:
      containers:
        - name: ratings
          image: istio/examples-bookinfo-ratings-v1
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 9080
Istio Sample App

$ istioctl create -f apps/bookinfo/route-rule-reviews-v2-v3.yaml

type: route-rule
spec:
  name: reviews-default
  destination: reviews.default.svc.cluster.local
  precedence: 1
  route:
    - tags:
        version: v2
        weight: 80
    - tags:
        version: v3
        weight: 20
## Istio K8s Services

```
> kubectl get services
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>CLUSTER-IP</th>
<th>EXTERNAL-IP</th>
<th>PORT(S)</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>details</td>
<td>10.0.0.31</td>
<td>&lt;none&gt;</td>
<td>9080/TCP</td>
<td>6m</td>
</tr>
<tr>
<td>kubernetes</td>
<td>10.0.0.1</td>
<td>&lt;none&gt;</td>
<td>443/TCP</td>
<td>7d</td>
</tr>
<tr>
<td>productpage</td>
<td>10.0.0.120</td>
<td>&lt;none&gt;</td>
<td>9080/TCP</td>
<td>6m</td>
</tr>
<tr>
<td>ratings</td>
<td>10.0.0.15</td>
<td>&lt;none&gt;</td>
<td>9080/TCP</td>
<td>6m</td>
</tr>
<tr>
<td>reviews</td>
<td>10.0.0.170</td>
<td>&lt;none&gt;</td>
<td>9080/TCP</td>
<td>6m</td>
</tr>
</tbody>
</table>
## Istio K8s App Pods

```
> kubectl get pods

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>details-v1-1520924117</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
<tr>
<td>productpage-v1-560495357</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
<tr>
<td>ratings-v1-734492171</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
<tr>
<td>reviews-v1-874083890</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
<tr>
<td>reviews-v2-1343845940</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
<tr>
<td>reviews-v3-1813607990</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>6m</td>
</tr>
</tbody>
</table>
```
Istio K8s System Pods

> kubectl get pods -n istio-system

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>istio-ca-797dfb66c5</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>istio-ingress-84f75844c4</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>istio-egress-29a16321d3</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>istio-mixer-9bf85fc68</td>
<td>3/3</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>istio-pilot-575679c565</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>grafana-182346ba12</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
<tr>
<td>prometheus-837521fe34</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>2m</td>
</tr>
</tbody>
</table>
Talk Agenda

✔ Istio Overview
☒ Service Level Objectives
☒ RED Dashboard
☒ Histogram Telemetry
☒ Istio Metrics Adapter
☒ Asking the Right Questions
Service Level Objectives

- **SLI** - Service Level Indicator
- **SLO** - Service Level Objective
- **SLA** - Service Level Agreement
Service Level Objectives
“SLIs drive SLOs which inform SLAs”

SLI - Service Level Indicator, a measure of the service that can be quantified

“95th percentile latency of homepage requests over past 5 minutes < 300ms”

Excerpted from “SLIs, SLOs, SLAs, oh my!” @sethvargo @lizthegrey

https://youtu.be/tEylFyxbDLE
“SLIs drive SLOs which inform SLAs”

**SLO** - Service Level Objective, a target for Service Level Indicators

“95th percentile homepage SLI will succeed 99.9% over trailing year”

Excerpted from
“SLIs, SLOs, SLAs, oh my!”
[@sethvargo @lizthegrey](https://youtu.be/tEylFyxbDLE)
“SLIs drive SLOs which inform SLAs”

**SLA** - Service Level Agreement, a legal agreement between a customer and a service provider based on SLOs

“Service credits if 95th percentile homepage SLI succeeds less than 99.5% over trailing year”

Excerpted from “SLIs, SLOs, SLAs, oh my!”

@sethvargo @lizthegrey

https://youtu.be/tEylFyxbDLE
Talk Agenda

- Istio Overview
- Service Level Objectives
- RED Dashboard
- Histogram Telemetry Collection
- Istio Metrics Adapter
- Asking the Right Questions
Emerging Standards

● **USE**
  ○ Utilization, Saturation, Errors
  ○ Introduced by Brendan Gregg [@brendangregg](https://twitter.com/brendangregg)
  ○ KPIs for host based health

● **The Four Golden Signals**
  ○ Latency, Traffic, Errors, Saturation
  ○ Covered in the Google SRE Book
  ○ Extended version of RED

● **RED**
  ○ Rate, Errors, Duration
  ○ Introduced by Tom Wilkie [@tom_wilkie](https://twitter.com/tom_wilkie)
  ○ KPIs for API based health, SLI focused
Containers?

- Ephemeral
- High Cardinality
- Difficult to Instrument
- Instrument Services, Not Containers
Istio Mixer Provided Telemetry

- Request Count by Response Code
- Request Duration
- Request Size
- Response Size
- Connection Received Bytes
- Connection Sent Bytes
- Connection Duration
- Template Based MetaData (Metric Tags)
RED

- **Rate**
  - Requests per second
  - First derivative of request count provided by Istio

- **Errors**
  - Unsuccessful requests per second
  - First derivative of failed request count provided by Istio

- **Duration**
  - Request latency provided by Istio
Rate

Request Volume

125 ops
100 ops
75 ops
50 ops
25 ops
0 ops

0 ops 10:57 10:58 10:59 11:00 11:01

- All
- 200s
Duration

Response Time by Source and Version

- `productpage.default-v1 -> v1 (p50)`
- `productpage.default-v1 -> v1 (p90)`
- `productpage.default-v1 -> v1 (p95)`
- `productpage.default-v1 -> v1 (p99)`
Duration

Problems:

● Percentiles > averages, but have limitations
  ○ Aggregated metric, fixed time window
  ○ Cannot be re-aggregated for cluster health
  ○ Cannot be averaged (common mistake)
● Stored aggregates are outputs, not inputs
● Difficult to measure cluster SLIs
● Leave a lot to be desired for
WE CAN DO BETTER
WE HAVE THE TECHNOLOGY
MATH

https://youtu.be/yCX1Ze3OcKo
Talk Agenda

- ✅ Istio Overview
- ✅ Service Level Objectives
- ✅ RED Dashboard
- ❏ Histogram Telemetry
- ❏ Istio Metrics Adapter
- ❏ Asking the Right Questions
Histogram

https://github.com/circonus-labs/circonusllhist
Log linear histogram

90 bins

Bin size increase by 10x

https://github.com/circonus-labs/circonusllhist
Duration - Histogram

https://github.com/circonus-labs/circonusllhist
Duration - SLI

SLI - “90th percentile latency of requests over past 5 minutes < 1,000ms”
Duration - Modes
Duration - Modes
Duration - Heatmap
Duration - Heatmap

Original Graph

May 14 2018, 19:10 (1M)

- API Requests (histogram) (ms)
  - 0.17s, 2.501s, 2.899s, 2.903s

- CAQL 2 (ms)
  - [16-17]: 132 of 2903 samples (78%, 5%, 24%)
This dashboard implements graphs for Rate, Duration, and Error service metrics from the IRONdb datasource.
RED - SLI Alerting
Talk Agenda

- ✔ Istio Overview
- ✔ Service Level Objectives
- ✔ RED Dashboard
- ✔ Histogram Telemetry
- ❑ Istio Metrics Adapter
- ❑ Asking the Right Questions
Istio Metrics Adapter

- Golang based adapter API
- In process (built into the Mixer executable)
  - Out of process for new adapter dev
- Set of handler hooks and YAML files
Istio Metrics Adapter

“SHOW ME THE CODE”

https://github.com/istio/istio/

https://github.com/istio/istio/blob/master/mixer/adapter/circonus
Istio Metrics Adapter

- guptasu and geeknoid: Reference new types from policy/v1beta1 (#5587)
- File: config, 3 days ago
- File: operatorconfig, 3 months ago
- File: circonus.go, 1 month ago
- File: circonus_test.go, 6 months ago
# Istio Metrics Adapter

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>bypass</td>
<td>Add a bypass adapter to Mixer for using gRPC backends via inline mode...</td>
<td>a day ago</td>
</tr>
<tr>
<td>circonus</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>cloudwatch</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>denier</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>dogstatsd</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>fluentd</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>kubernetesenv</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>list</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>memquota</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>noop</td>
<td>Bump up coverage numbers in a few packages. (#4519)</td>
<td>2 months ago</td>
</tr>
<tr>
<td>opa</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>prometheus</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>rbac</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>redisquota</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>servicecontrol</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>solarwinds</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>stackdriver</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>statsd</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
<tr>
<td>stdio</td>
<td>Doc updates (#5914)</td>
<td>4 days ago</td>
</tr>
</tbody>
</table>
// HandleMetric submits metrics to Circonus via circonus-gometrics
func (h *handler) HandleMetric(ctx context.Context, insts []*metric.Instance) error {

    for _, inst := range insts {
        metricName := inst.Name
        metricType := h.metrics[metricName]

        switch metricType {

        case config.GAUGE:
            value, _ := inst.Value.(int64)
            h.cm.Gauge(metricName, value)

        case config.COUNTER:
            h.cm.Increment(metricName)

        }
    }
}
Istio Metrics Adapter

case config.DISTRIBUTION:
    value, _ := inst.Value.(time.Duration)
    h.cm.Timing(metricNameName, float64(value))
}

}

return nil
}
Istio Metrics Adapter

handler struct {
    cm    *cgm.CirconusMetrics
    env   adapter.Env
    metrics map[string]config.Params_MetricInfo_Type
    cancel context.CancelFunc
}
And some YAML

```yaml
metrics:
- name: requestcount.metric.istio-system
type: COUNTER
- name: requestduration.metric.istio-system
type: DISTRIBUTION
- name: requestsize.metric.istio-system
type: GAUGE
- name: responsesize.metric.istio-system
type: GAUGE
```
Buffer metrics, then report

env.ScheduleDaemon(
    func() {

        ticker := time.NewTicker(b.adpCfg.SubmissionInterval)
        for {
            select {
                case <-ticker.C:
                    cm.Flush()
                case <-adapterContext.Done():
                    ticker.Stop()
                    cm.Flush()
                    return
            }
        }
    }
)
Talk Agenda

- Istio Overview
- Service Level Objectives
- RED Dashboard
- Histogram Telemetry
- Istio Metrics Adapter
- Asking the Right Questions
Your boss wants to know

● How many users got angry on the Tuesday slowdown after the big marketing promotion?
● Are we over-provisioned or under-provisioned on our purchasing checkout service?
● Other business centric questions
The Slowdown

- Marketing launched a new product
- Users complained the site was slow
- Median human reaction time is 215 ms [1]
- If users get angry (rage clicks) when requests take more than 500 ms, how many users got angry?

The Slowdown

1. Record all service request latencies as distribution
2. Plot as a heatmap
3. Calculate percentage of requests that exceed 500ms SLI using inverse percentiles
4. Multiply result by total number requests, integrate over time
The Slowdown

4 million slow requests
Under or Over Provisioned?

- “It depends”
- Time of day, day of week
- Special events
- Behavior under load
- Latency bands shed some light
Conclusions

- Monitor services, not containers
- Record distributions, not aggregates
- Istio gives you RED metrics for free
- Use math to ask the right questions
Thank you! Questions?

Tweet me @phredmoyer