
KubeCon
CloudNativeCon
Europe 2020
Virtual
HELM
KEEP CLOUD NATIVE
CONNECTED


KubeCon | CloudNativeCon

Europe 2020

Virtual

Kubernetes SIG Instrumentation

Instrumenting for Day Two Concerns

Marek Siarkowicz, Google
@serathius

Plan

- Charter
- Projects
- Ongoing efforts
- Where you can find us?

KubeCon | CloudNativeCon

Europe 2020

Virtual

Kubernetes SIG Instrumentation

Charter

Mission statement

Best practices for cluster observability through metrics and logging across all Kubernetes components

https://github.com/kubernetes/community/tree/master/sig-instrumentation

In Scope

- Revolve around the **process of instrumenting**
- **Advise** contributors on instrumentation decisions
- **Coordinate instrumentation requirements** through finding common APIs

Out of Scope

- **Processing of signals**. For example ingesting metrics, logs, events into external systems.
- **Cloud provider specific** addons are out of scope and should be taken care of by the respective SIG.

Team

- 19 members
- 2 Chairs
  - Han Kang @logicalhan
  - Elana Hashman @ehashman

- 2 Tech Leads
  - David Ashpole @dashpole
  - Frederic Branczyk @brancz

KubeCon | CloudNativeCon

Europe 2020

Virtual

Kubernetes SIG Instrumentation

Projects

Projects

- Metrics API
- Metrics Server
- Custom Metrics Apiserver
- Kube State Metrics

Metrics API (part 1)

Generic API exposing metrics that can be consumed by Kubernetes built in **autoscaling pipeline**

https://github.com/kubernetes/metrics

Metrics API (part 2)

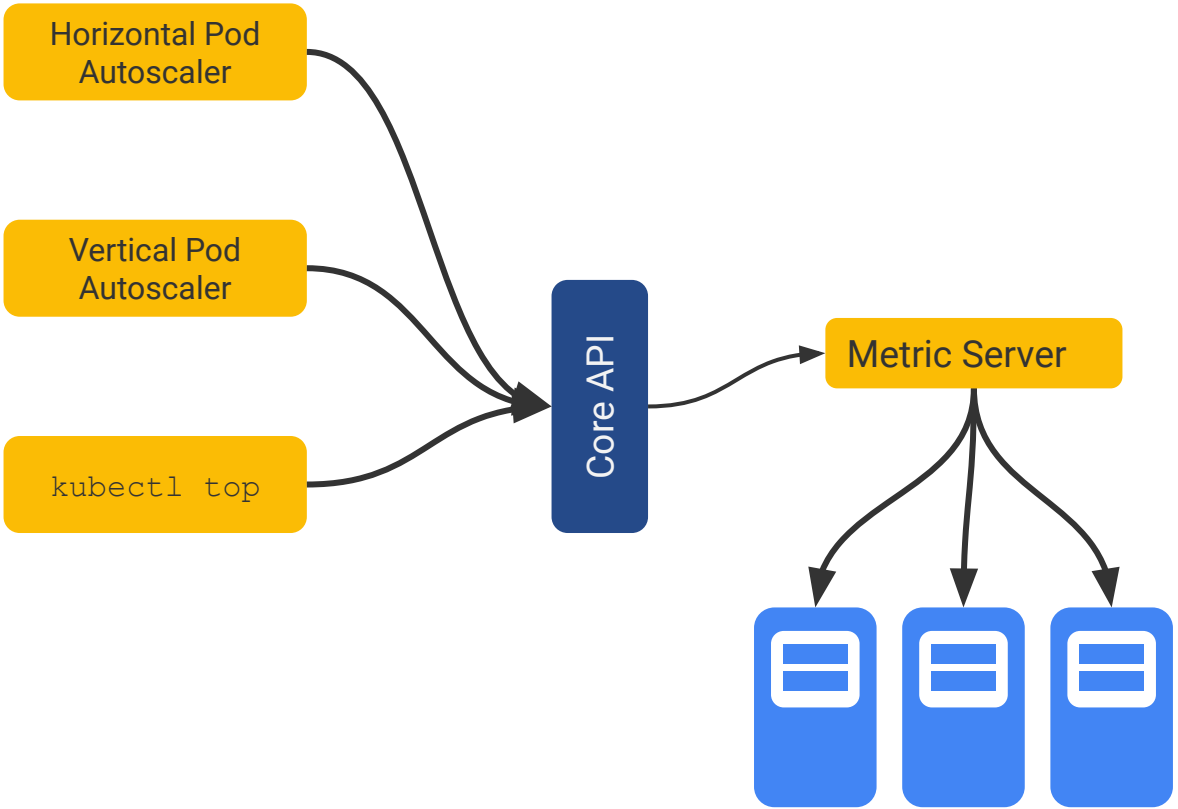
Core Metrics - CPU, Memory

Custom Metrics - e.g. requests per second to pod

External Metrics - e.g. length of external pubsub queue

https://github.com/kubernetes/metrics

Core Metrics Pipeline

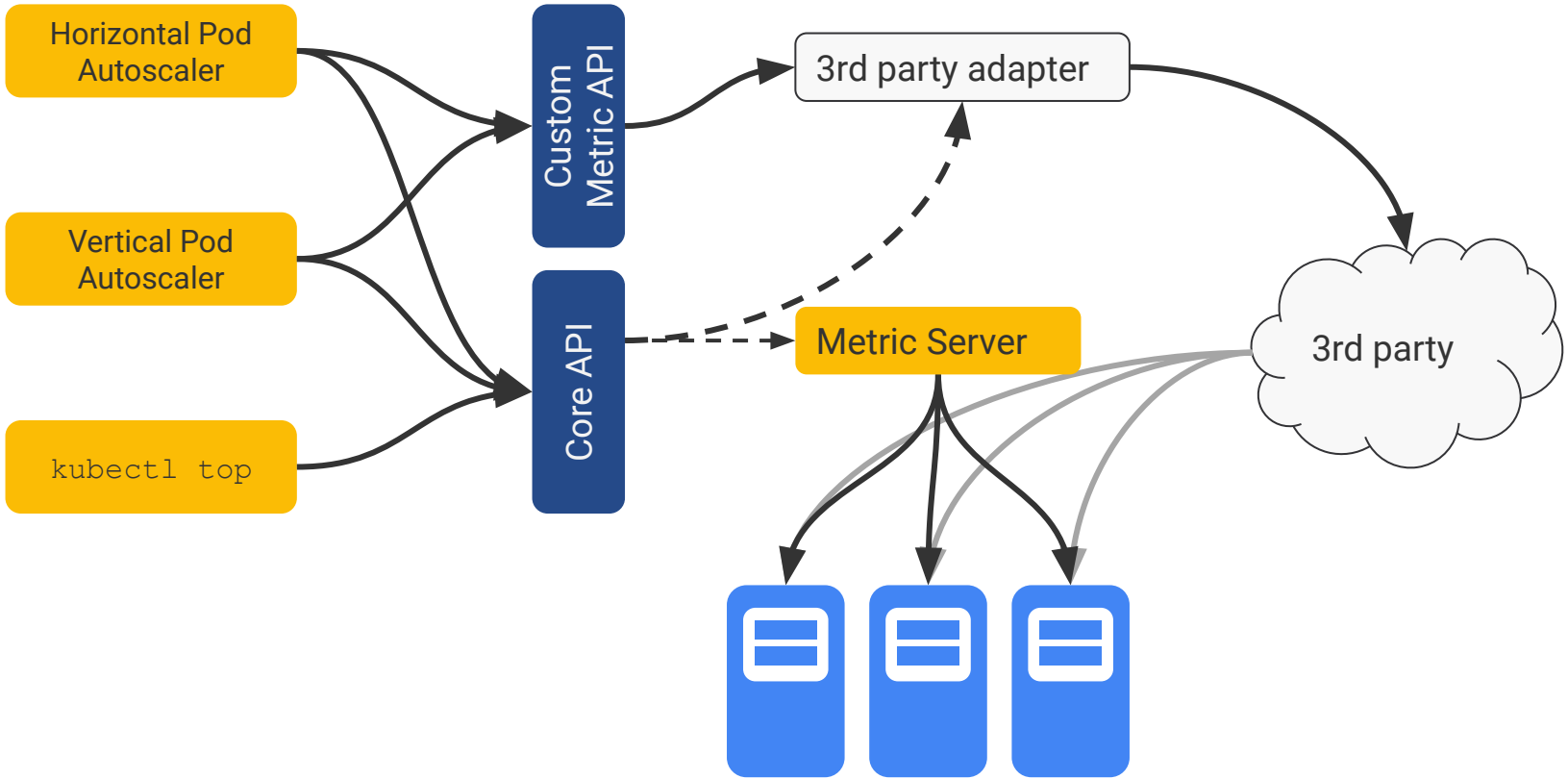


Metrics Server

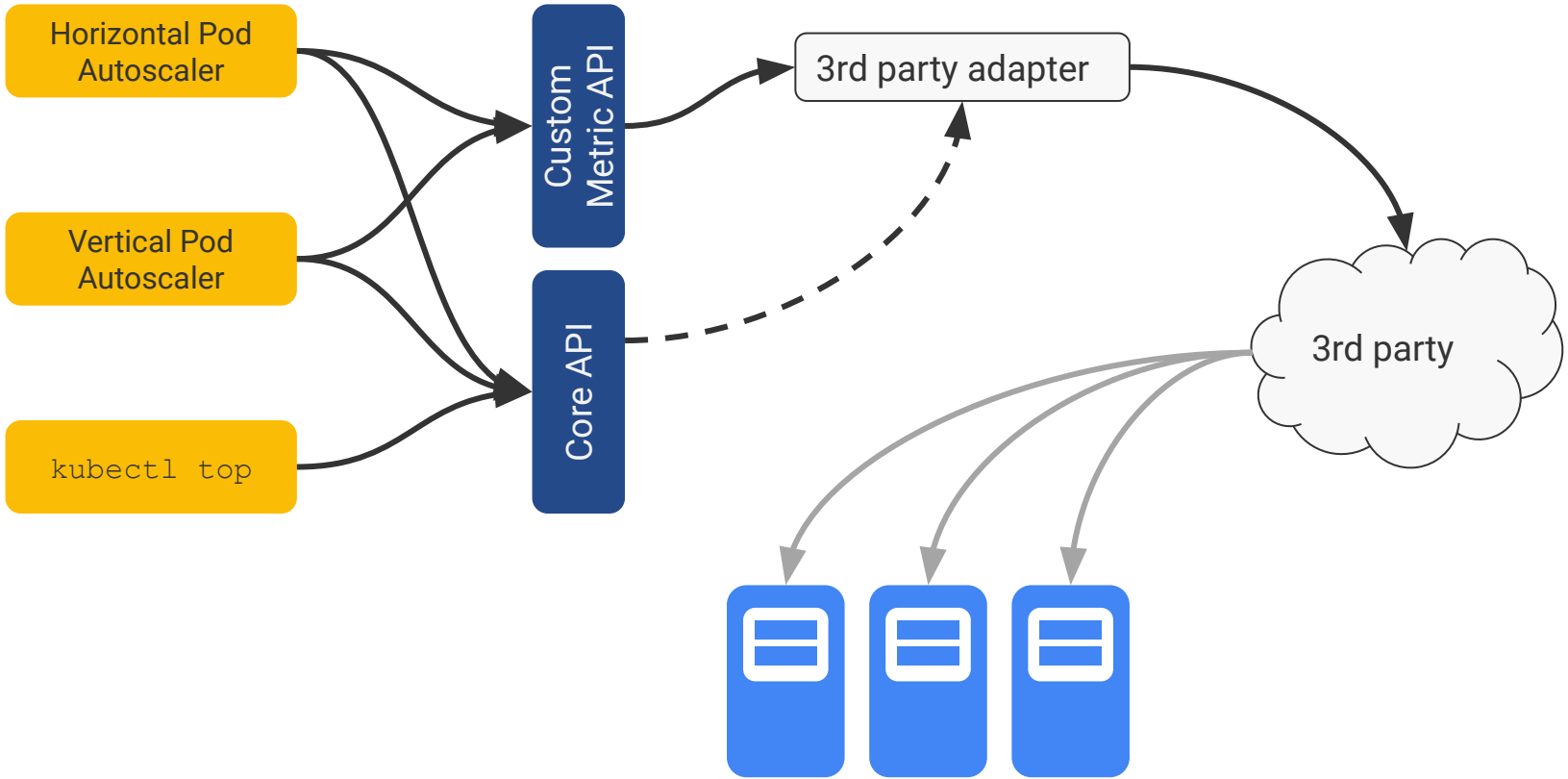
- Concrete implementation of Core Metrics API
- Doesn’t require full blown monitoring system
- Supports up to 5,000 node clusters
- Stores metric values in memory.

https://github.com/kubernetes-sigs/metrics-server

Custom/External Metrics Pipeline



Custom/External Metrics Pipeline



Custom Metrics Apiserver

Library with boilerplate code for setting up an adapter serving Custom/External Metrics API.

Example consumers:

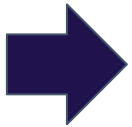
- K8s-prometheus-adapter
- Datadog-agent
- Custom-metrics-stackdriver-adapter

https://github.com/kubernetes-sigs/custom-metrics-apiserver

Kube state metrics

- Generates metrics from state of Kubernetes objects
- Metrics exposed in Prometheus format

```
apiVersion: apps/v1
kind: Deployment
metadata:
  Name: app1
spec:
  ...
  replicas: 3
status:
  ...
  replicas: 2
```



```
kube_deployment_spec_replicas{name="app1"} 3

kube_deployment_status_replicas{name="app1"} 2
```

https://github.com/kubernetes/kube-state-metrics

KubeCon | CloudNativeCon

Europe 2020

Virtual

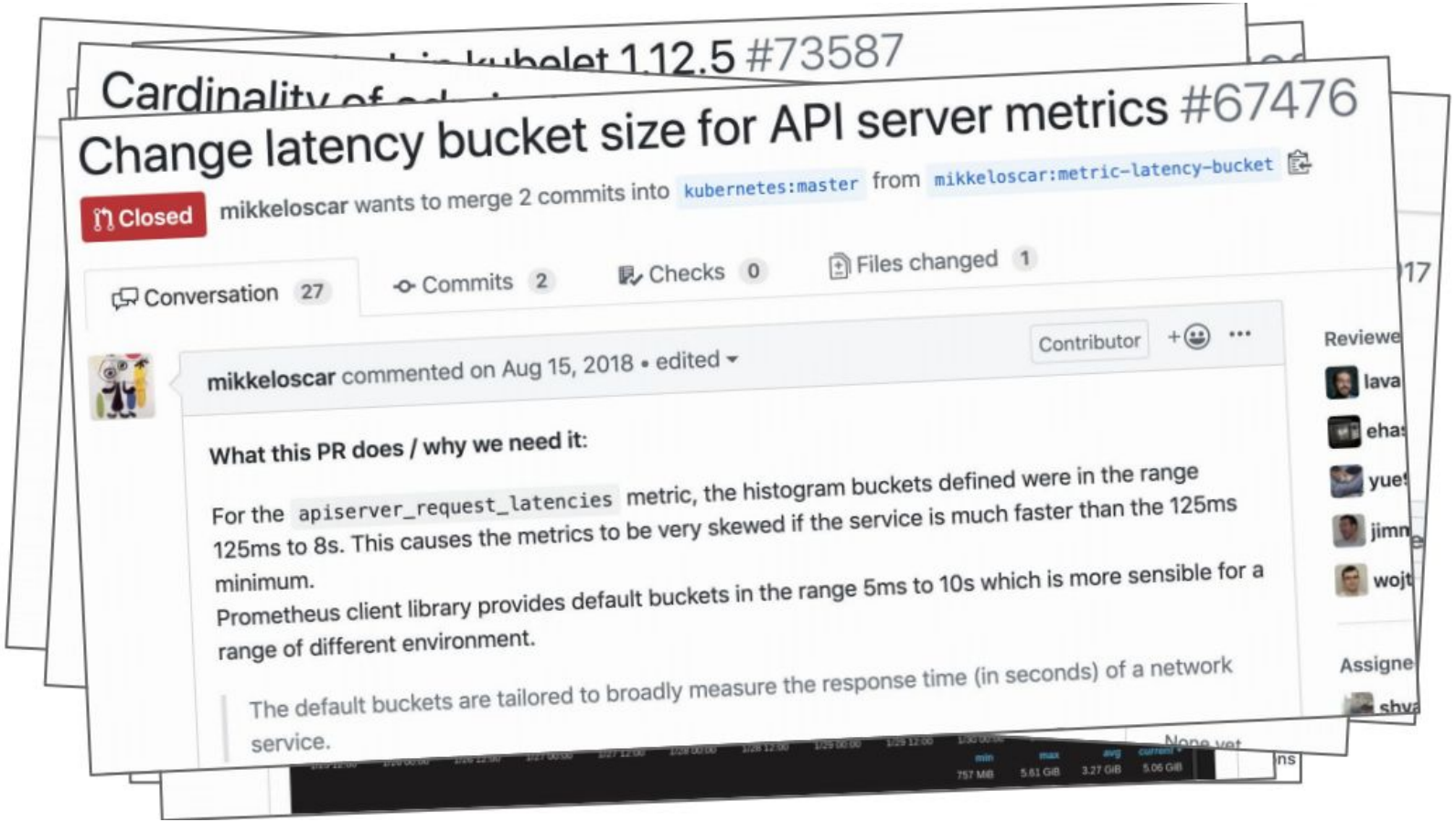
Kubernetes SIG Instrumentation

Ongoing efforts

Ongoing efforts

- Monitoring
- Logging
- Tracing

Monitoring


Cardinality of
kubelet 1.12.5 #73587
Change latency bucket size for API server metrics #67476
Closed
mikkeloscar wants to merge 2 commits into kubernetes:master from mikkeloscar:metric-latency-bucket
Conversation 27
Commits 2
Checks 0
Files changed 1
mikkeloscar commented on Aug 15, 2018 • edited
Contributor
What this PR does / why we need it:
For the apiserver_request_latencies metric, the histogram buckets defined were in the range 125ms to 8s. This causes the metrics to be very skewed if the service is much faster than the 125ms minimum.
Prometheus client library provides default buckets in the range 5ms to 10s which is more sensible for a range of different environment.
The default buckets are tailored to broadly measure the response time (in seconds) of a network service.
Reviewe
lava
eha
yue
jimn
wojt
Assigne
shva
None yet
min
max
avg
current
757 MiB
5.61 GiB
3.27 GiB
5.06 GiB

Metrics Overhaul - GA

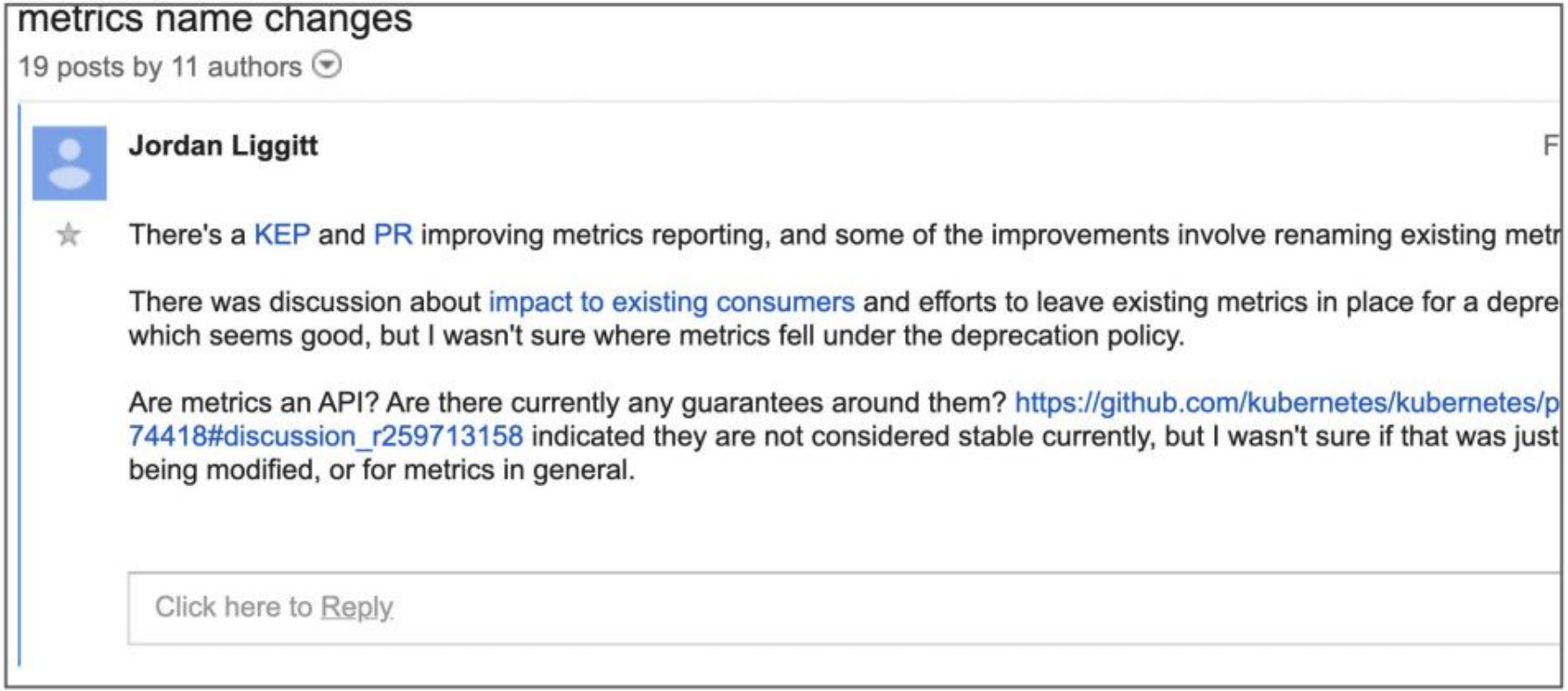
Goals

- Bring existing metrics in-line with guidelines
- Fix known issues with existing metrics

https://github.com/kubernetes/enhancements/blob/master/keps/sig-instrumentation/20181106-kubernetes-metrics-overhaul.md

Are metrics an API?

KubeCon CloudNativeCon Europe 2020 Virtual



metrics name changes

19 posts by 11 authors

Jordan Liggitt F

There's a KEP and PR improving metrics reporting, and some of the improvements involve renaming existing metr

There was discussion about impact to existing consumers and efforts to leave existing metrics in place for a depre which seems good, but I wasn't sure where metrics fell under the deprecation policy.

Are metrics an API? Are there currently any guarantees around them? https://github.com/kubernetes/kubernetes/p 74418#discussion_r259713158 indicated they are not considered stable currently, but I wasn't sure if that was just being modified, or for metrics in general.

Click here to Reply

Metrics Stabilization - Beta

- Stability Classes
  - Alpha - no guarantees
  - Stable - adheres to deprecation policy
- Deprecation
  - Stable (v1.18) -> Deprecated (v1.19) -> Hidden (v1.20) -> Deleted (v1.21)

Metrics Framework

KubeCon CloudNativeCon Europe 2020 Virtual

Before

```
import "github.com/prometheus/client_golang/prometheus"

var (
    someCounter = prometheus.CounterOpts{
        Name: "some_counter",
        Help: "it counts",
    }
 )
```

```
# HELP some_counter it counts
# TYPE some_counter counter
some_counter 0
```

After

```
import "k8s.io/component-base/metrics"

var (
    someCounter = metrics.CounterOpts{
        Name: "some_counter",
        Help: "it counts",
        StabilityLevel: metrics.STABLE,
        DeprecatedVersion: "1.18",
    }
 )
```

```
# HELP some_counter (Deprecated from 1.18) it counts
# TYPE some_counter counter
some_counter 0
```

Logging

Before

```
klog.Infof("%s %s: (%v) %v%v%v [%s %s]", rl.req.Method, rl.req.RequestURI, latency,
rl.status, rl.statusStack, rl.addedInfo, rl.req.UserAgent(), rl.req.RemoteAddr)
```

https://github.com/kubernetes/enhancements/tree/master/keps/sig-instrumentation/1602-structured-logging

Structured Logging - Alpha

- Introduce structured functions to klog
- Initiate migration
  - Change 99% of log volume by updating 22 log calls
- Introduce alternative logging formats
  - --logging-format flag
  - JSON format

https://github.com/kubernetes/enhancements/tree/master/keps/sig-instrumentation/1602-structured-logging

Structured Logging - Alpha

Before

```
klog.Infof("%s %s: (%v) %v%v%v [%s %s]", rl.req.Method,
rl.req.RequestURI, latency, rl.status, rl.statusStack,
rl.addedInfo, rl.req.UserAgent(), rl.req.RemoteAddr)
```

```
I0619 16:07:20.559376 9160 http.go:79] GET /metrics:
(1.512ms) 200 [Mozilla/5.0 AppleWebKit/537.36
10.56.1.19:51756]
```

After

```
klog.InfoS("HTTP", "verb", rl.req.Method, "URI",
rl.req.RequestURI, "latency", latency, "resp", rl.status,
"userAgent", rl.req.UserAgent(), "srcIP",rl.req.RemoteAddr)
```

```
I0619 16:07:20.559376 9160 http.go:79] "HTTP" verb="GET"
URI="/metrics" latency="1.512ms" resp=200
userAgent="Mozilla/5.0 AppleWebKit/537.36" srcIP="127.0.0.1"
```

```
{
  "ts": 1580306777.04728,
  "msg": "HTTP",
  "verb": "GET",
  "URI": "/metrics",
  "latency": "1.512ms",
  "resp": 200,
  "userAgent": "Mozilla/5.0 AppleWebKit/537.36",
  "srcIP": "127.0.0.1”
}
```

Log Sanitization - Proposal

Kubernetes Security Audit results

- Bearer tokens are revealed in logs
- iSCSI volume storage cleartext secrets in logs

Suggestion by auditors

- Ensure that sensitive data cannot be trivially stored in logs

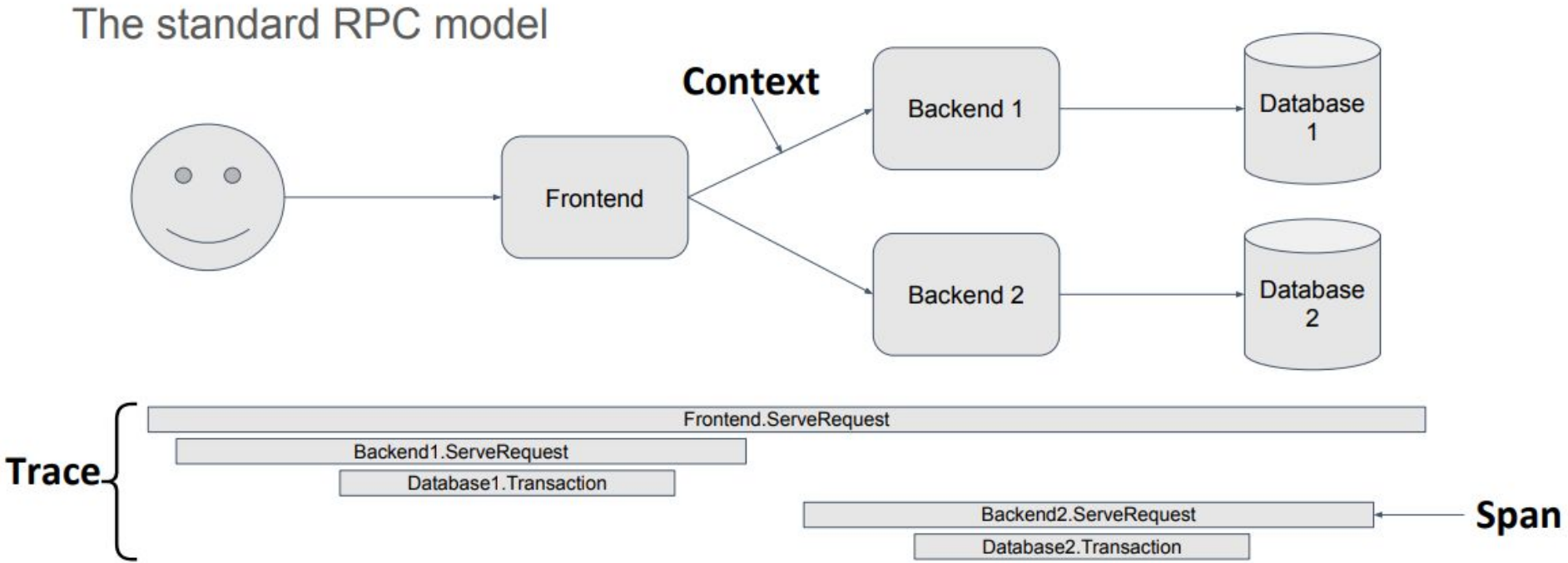
https://github.com/kubernetes/enhancements/pull/1754

Log Sanitization - Proposal

- Introduce a struct annotation for sensitive data
- Annotate Kubernetes codebase
- Introduce a sensitivity filter in logging library

https://github.com/kubernetes/enhancements/pull/1754

Tracing



https://github.com/kubernetes/enhancements/pull/1458

Tracing - Proposal

Goal

- Introduce tracing to kubernetes apiserver and clients

Future

- Tracing in kubernetes controllers
- Use traces for life cycle of Kubernetes Objects

https://github.com/kubernetes/enhancements/pull/1458

Where you can find us?

Where you can find us?

- Slack #sig-instrumentation
- Mailing list kubernetes-sig-instrumentation@googlegroups.com
- Regular biweekly meetings on Thursdays at 9:30 PT (Pacific Time)

Thank you