Grafana Labs

# Prometheus Sparse High-Resolution Histograms in Action

Ganesh Vernekar

KubeCon EU, May 19 2022

# About Me

Ganesh Vernekar

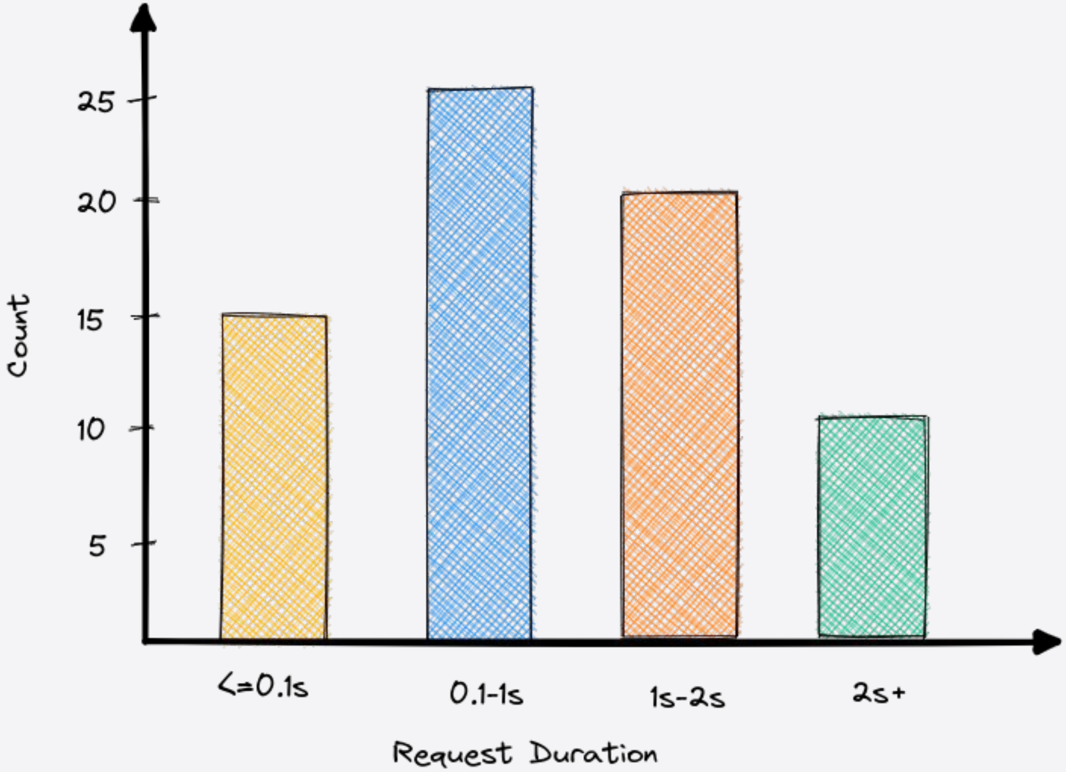
Senior Software Engineer at Grafana Labs

Prometheus Team Member

Prometheus TSDB Maintainer

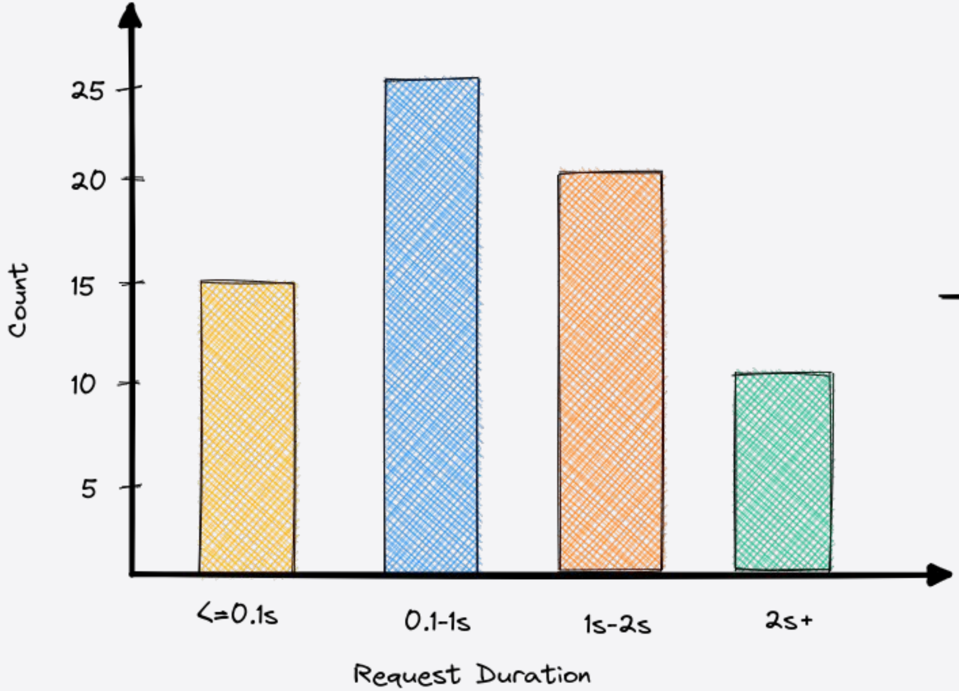
🐦 @_codesome

# What is a histogram?



@_codesome

# This is how it is stored in Prometheus



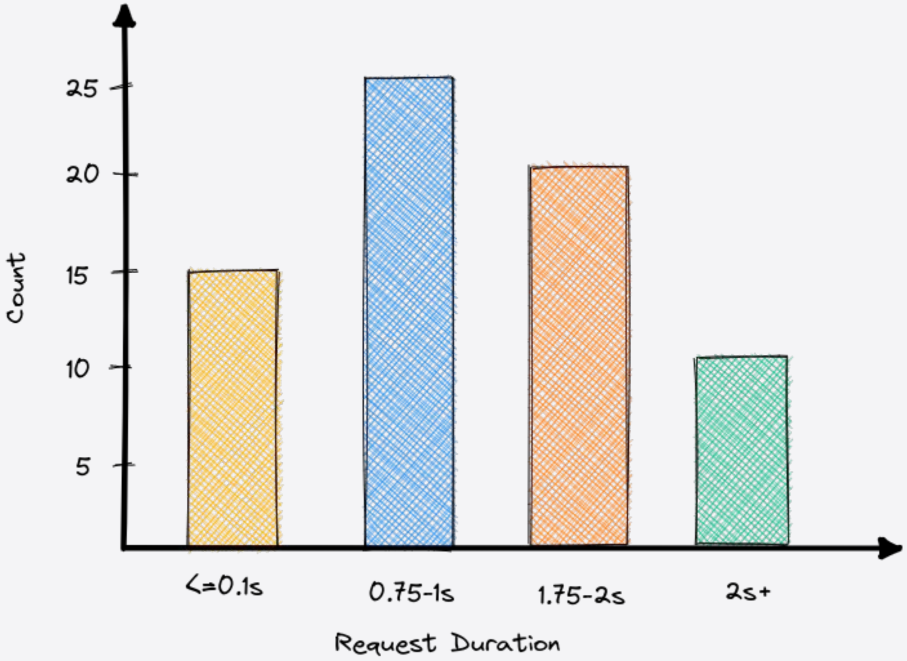
```
series_bucket{le="0.1"}  15
series_bucket{le="1.0"}  40
series_bucket{le="2.0"}  60
series_bucket{le="+Inf"} 70
series_count 70
series_sum 60.75
```

# Problem with this design

Pre-defined, cumulative, non-sparse, buckets.

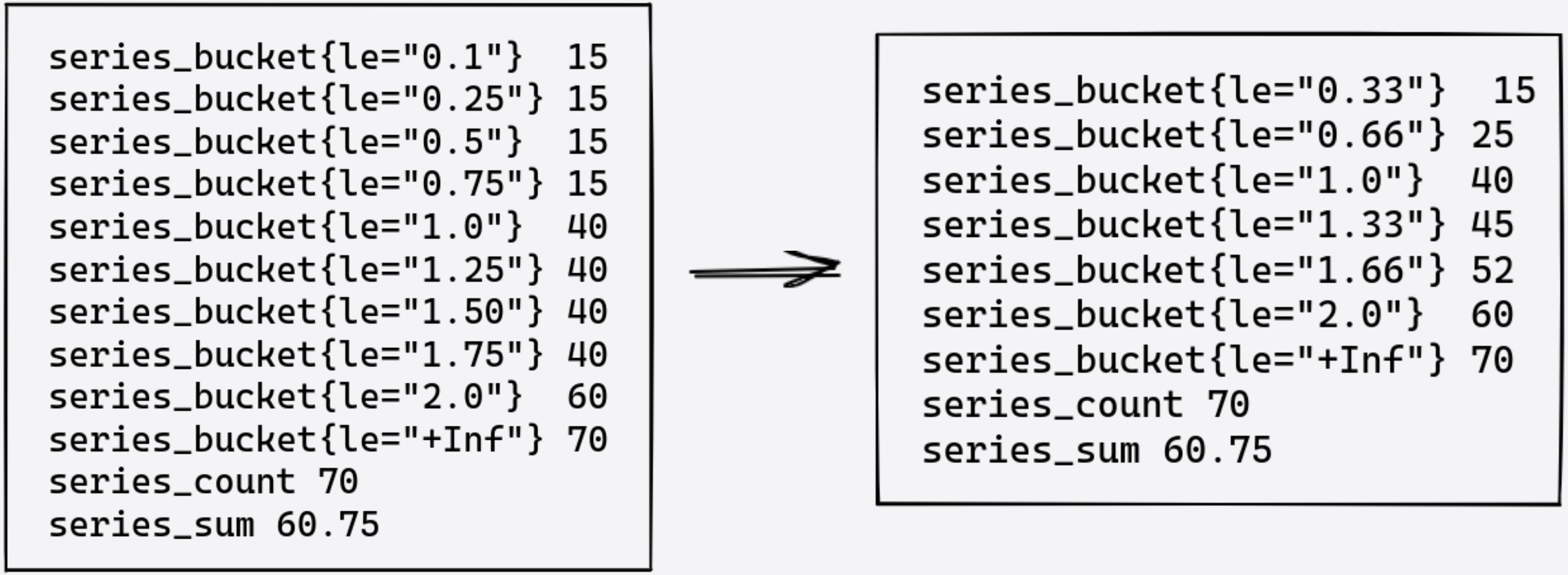
Got the buckets wrong? Re-instrument and re-deploy it everywhere

```
series_bucket{le="0.1"}  15
series_bucket{le="0.25"} 15
series_bucket{le="0.5"}  15
series_bucket{le="0.75"} 15
series_bucket{le="1.0"}  40
series_bucket{le="1.25"} 40
series_bucket{le="1.50"} 40
series_bucket{le="1.75"} 40
series_bucket{le="2.0"}  60
series_bucket{le="+Inf"} 70
series_count 70
series_sum 60.75
```



# Problem with this design

Changing bucket layout is painful, can’t correlate anymore in many cases

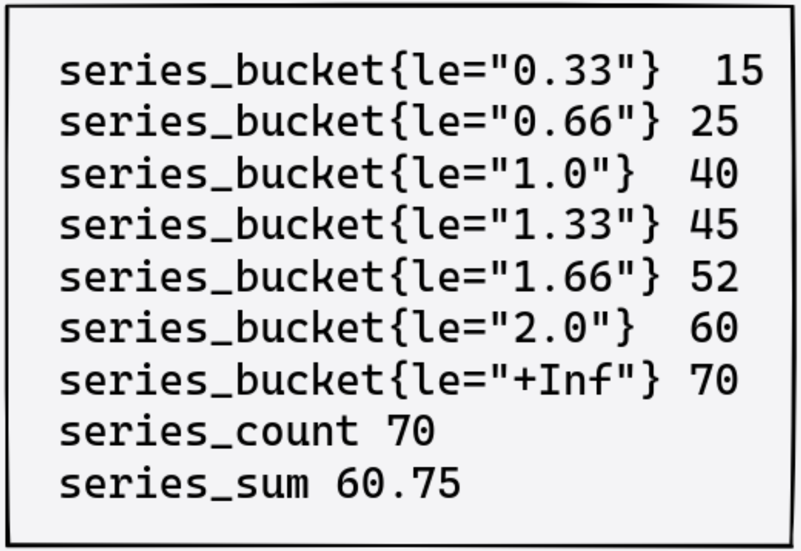


```
series_bucket{le="0.1"}  15
series_bucket{le="0.25"} 15
series_bucket{le="0.5"}  15
series_bucket{le="0.75"} 15
series_bucket{le="1.0"}  40
series_bucket{le="1.25"} 40
series_bucket{le="1.50"} 40
series_bucket{le="1.75"} 40
series_bucket{le="2.0"}  60
series_bucket{le="+Inf"} 70
series_count 70
series_sum 60.75
```

```
series_bucket{le="0.33"}  15
series_bucket{le="0.66"} 25
series_bucket{le="1.0"}  40
series_bucket{le="1.33"} 45
series_bucket{le="1.66"} 52
series_bucket{le="2.0"}  60
series_bucket{le="+Inf"} 70
series_count 70
series_sum 60.75
```

# Problem with this design

`total_series_per_histogram = num_buckets + 3`



```
series_bucket{le="0.33"}  15
series_bucket{le="0.66"} 25
series_bucket{le="1.0"}  40
series_bucket{le="1.33"} 45
series_bucket{le="1.66"} 52
series_bucket{le="2.0"}  60
series_bucket{le="+Inf"} 70
series_count 70
series_sum 60.75
```

# Here comes the sparse high resolution histograms

Let’s build it step by step

# Result of
# multi year study and research

## Prometheus Histograms – Past, Present, and Future

Björn “Beorn” Rabenstein
PromCon EU, Munich – 2019-11

Grafana Labs

## Secret History of Prometheus Histograms

Björn “Beorn” Rabenstei
FOSDEM, Brussels – 20

Grafana Labs

## Better Histograms for Promethe

Björn “Beorn” Rabenstein

Grafana Labs

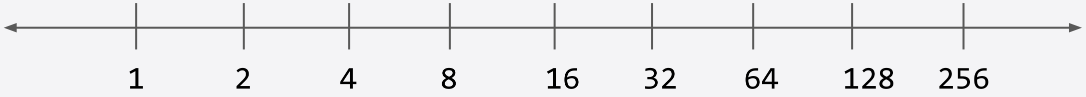
## A New Kid in Histogram Town

Björn “Beorn” Rabenstein

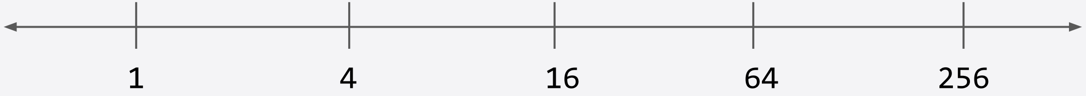
Grafana Labs

# First property: Fixed bucket boundaries

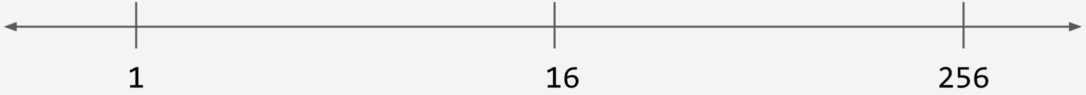
Factor of **$2^1$**



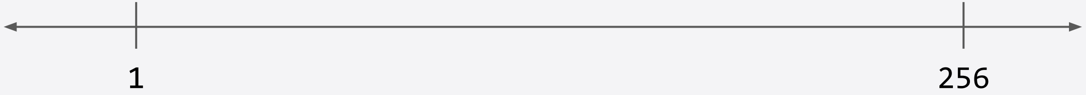
**$2^2$**, same as merging consecutive buckets from above



**$2^4$**

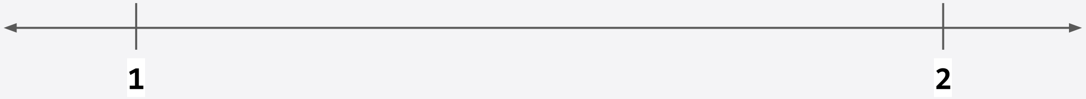


**$2^8$**

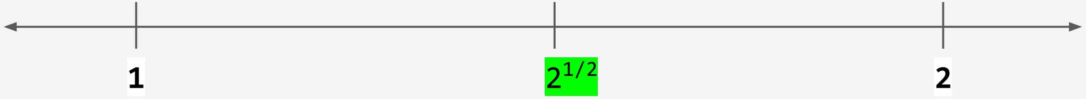


# Fixed bucket boundaries, higher resolution

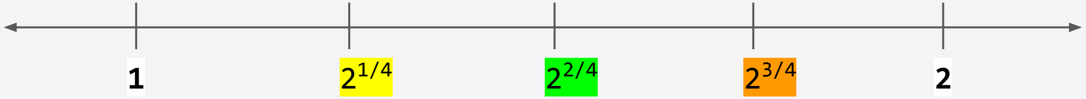
Factor of $\mathbf{2^1}$



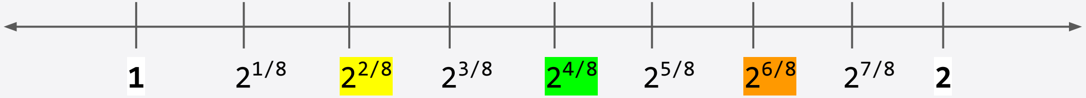
$\mathbf{2^{1/2}}$, gives 2 buckets between 1 and 2.



$\mathbf{2^{1/4}}$, gives 2 buckets between 1 ↔ $2^{1/2}$, and $2^{1/2}$↔2



$\mathbf{2^{1/8}}$, same as above, 2 buckets between above boundaries



# Fixed bucket boundaries

**$2^{1}$:**

…| 0.0625 | 0.125 | 0.25 | 0.5 | **1** | 2 | 4 | 8 | 16 |…

**$2^{2}$:**

…| 0.0625 | 0.25 | **1** | 4 | 16 | 64 |…

**$2^{1/2}$:**

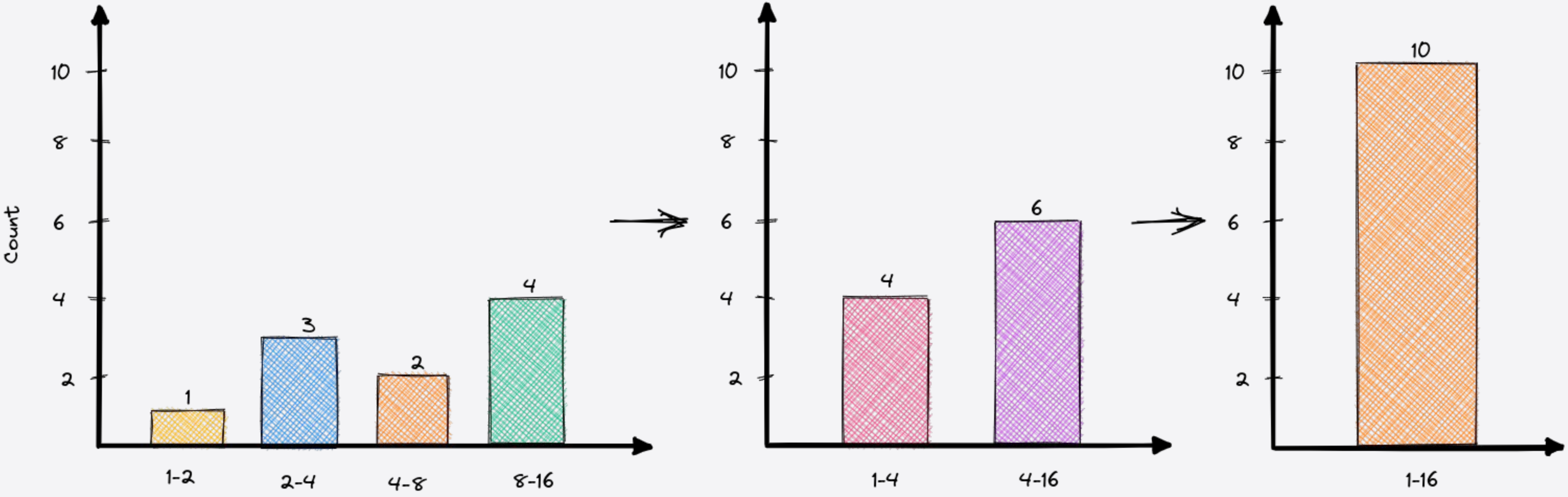
…| **0.25** | 0.5/$2^{1/2}$ | **0.5** | 1/$2^{1/2}$ | **1** | $2^{1/2}$ | **2** | 2*$2^{1/2}$ | **4** | 4*$2^{1/2}$ | **8** |…

**$2^{1/4}$:**

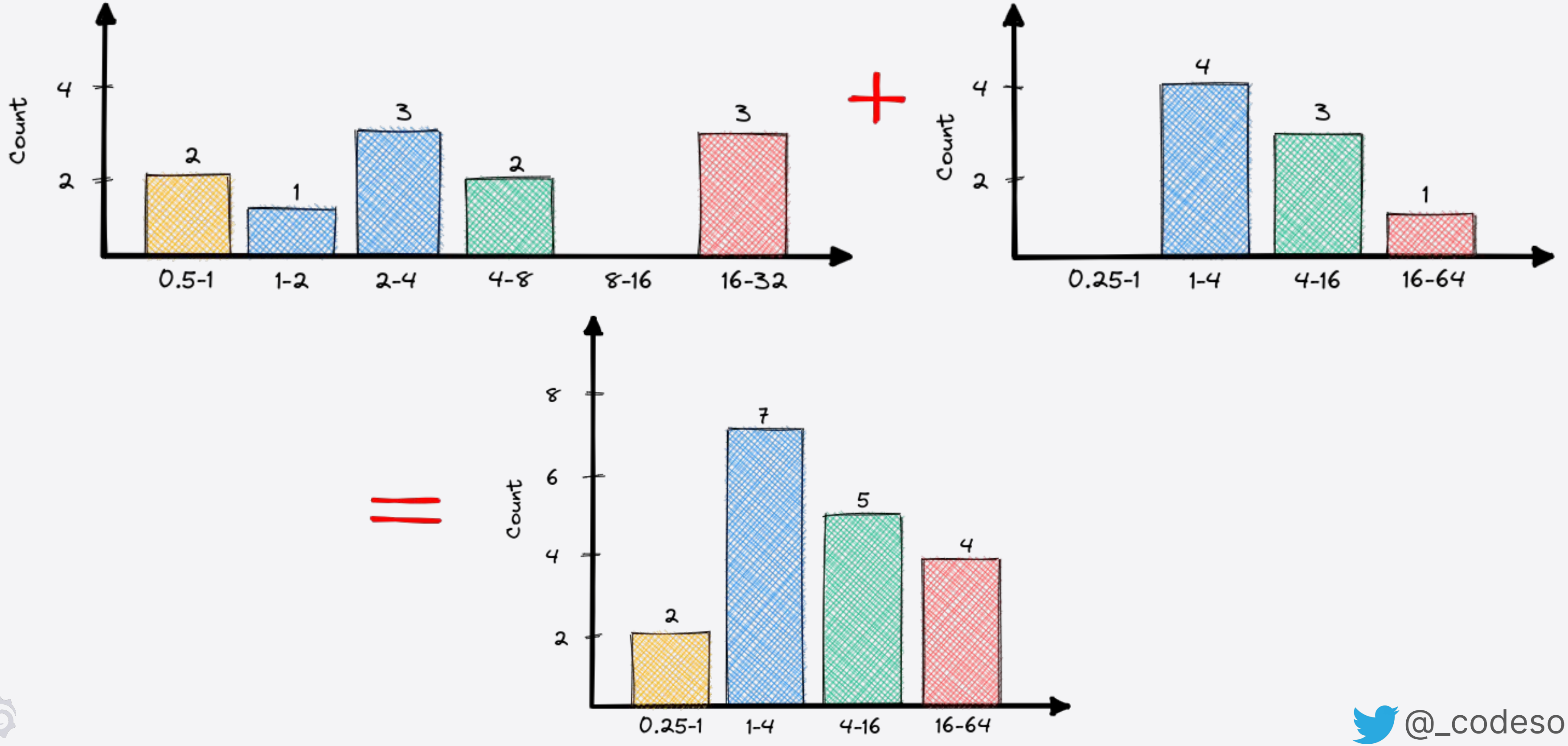
…| **0.5** | 1/$2^{3/4}$ | 1/$2^{2/4}$ | 1/$2^{1/4}$ | **1** | $2^{1/4}$ | $2^{2/4}$ | $2^{3/4}$ | **2** | 2*$2^{1/4}$ | 2*$2^{2/4}$ | 2*$2^{3/4}$ | **4** |…

# Why is it like this?

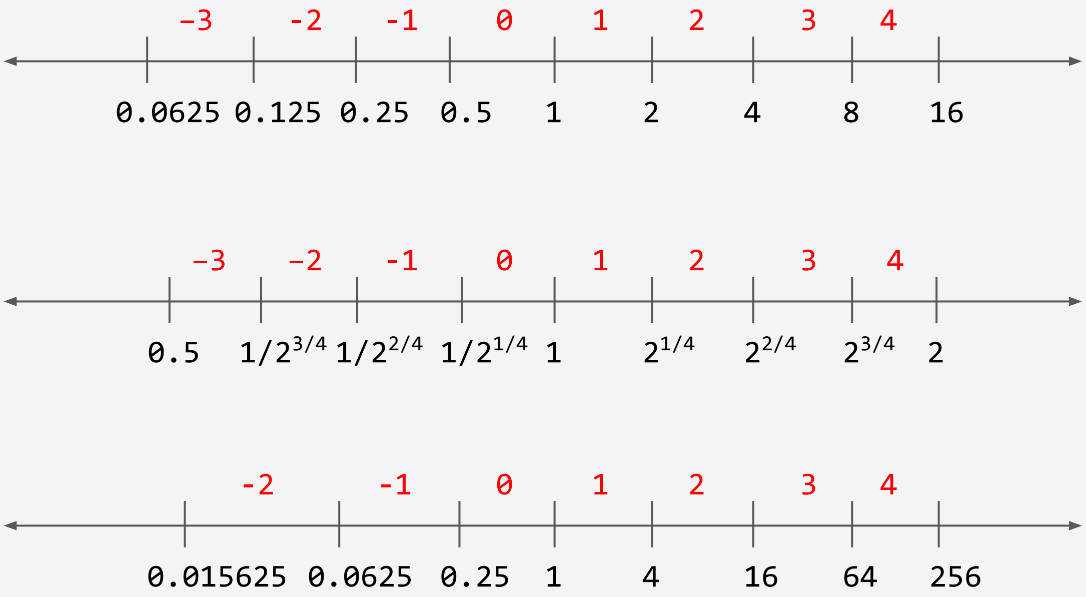
# Convert to a lower resolution



# Combine different resolutions

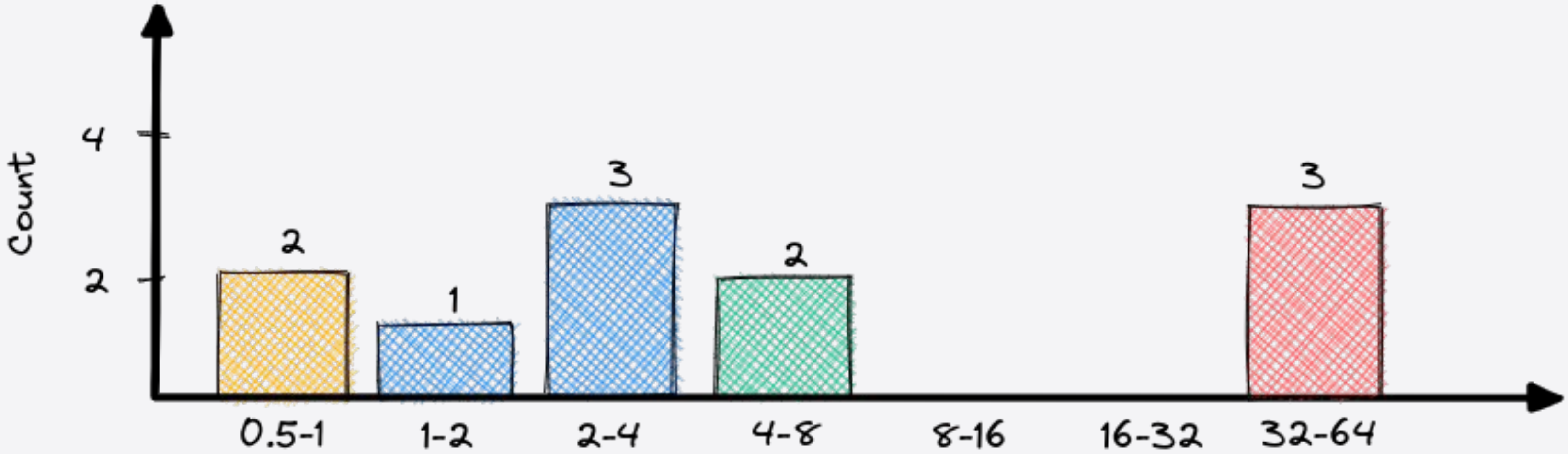


# Unique ID per bucket



@_codesome

# Finally...encoding these histograms



```
<metadata> // Resolution, sum, count, etc

[ (0, 4), (2, 1) ] // “Spans”, bucket layout

[ 2, 1, 3, 2, 3 ] // The bucket values
```

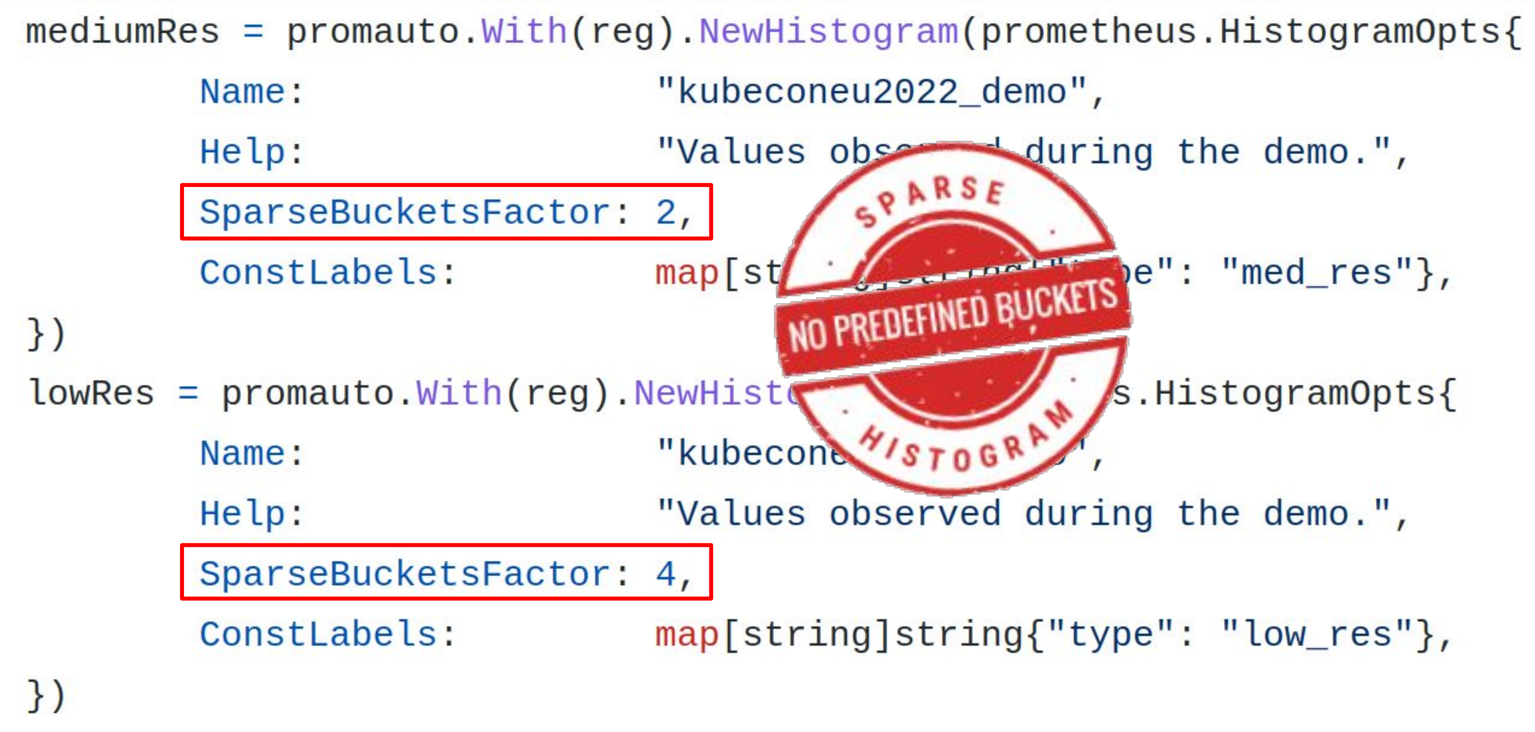
# One time series per histogram

```
my_histogram{foo=”bar”} =>

( T, <metadata> // Resolution, sum, count, etc
     [ (0, 4), (1, 1) ] // “Spans”, bucket layout
     [ 2, 1, 3, 2, 3 ] // The bucket values
)
```

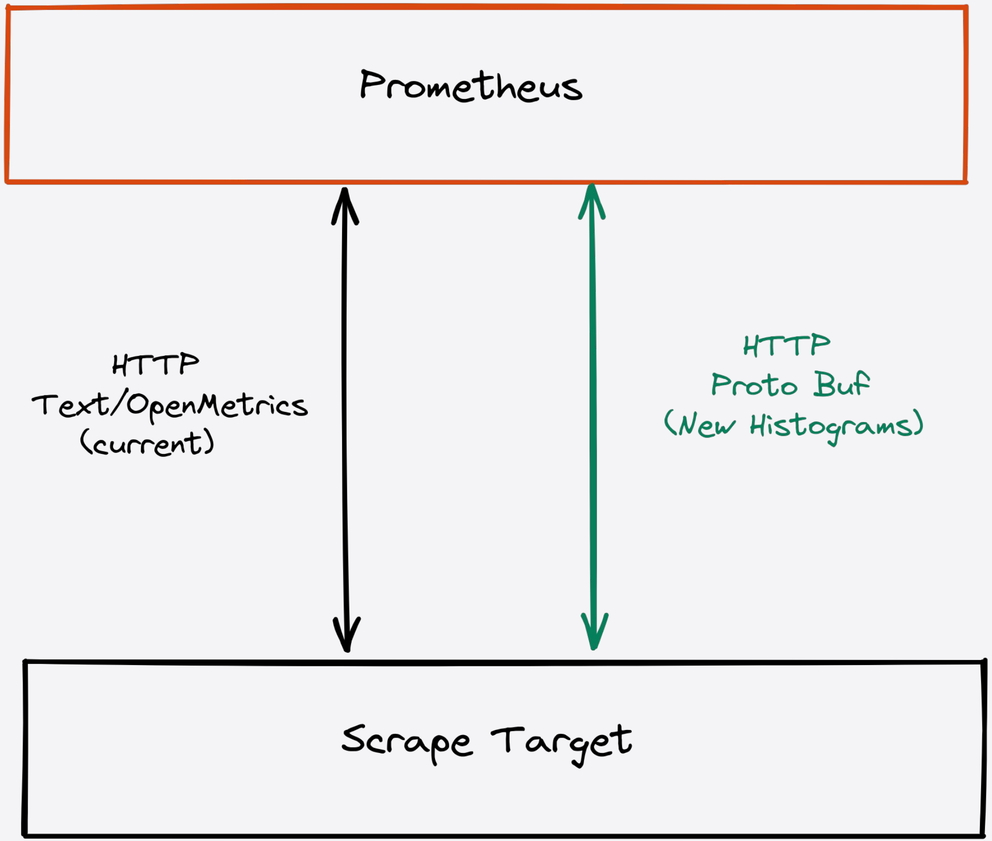
# How do I instrument this?

# Instrumentation

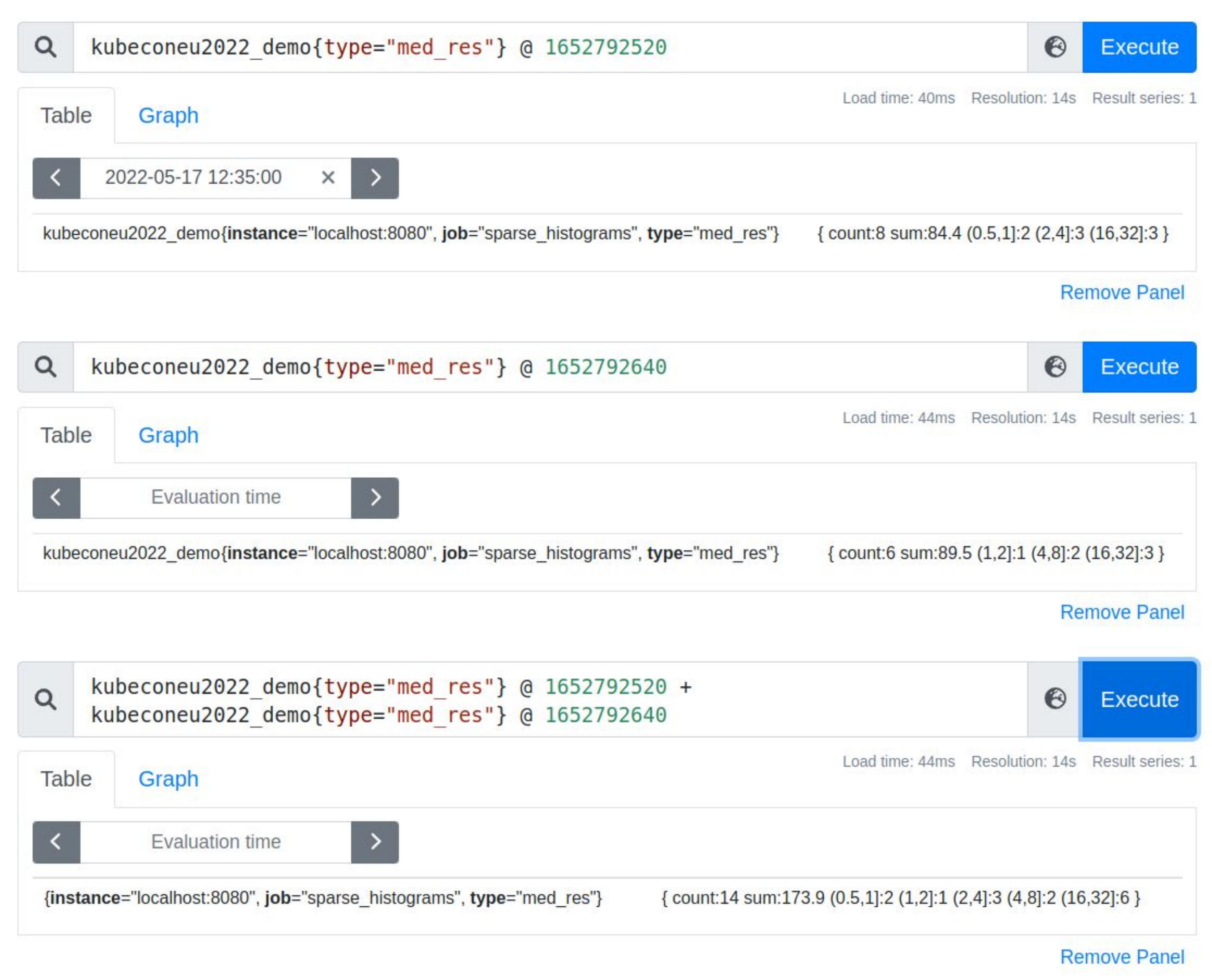


```
mediumRes = promauto.With(reg).NewHistogram(prometheus.HistogramOpts{
        Name:                "kubeconeu2022_demo",
        Help:                "Values obs          during the demo.",
        SparseBucketsFactor: 2,
        ConstLabels:         map[st                pe": "med_res"},
})
lowRes = promauto.With(reg).NewHist              s.HistogramOpts{
        Name:                "kubecone           ",
        Help:                "Values observed during the demo.",
        SparseBucketsFactor: 4,
        ConstLabels:         map[string]string{"type": "low_res"},
})
```

# Scraping in this PoC



# DEMO TIME!


Execute
Table
Graph
Load time: 40ms Resolution: 14s Result series: 1
Remove Panel
Execute
Table
Graph
Load time: 44ms Resolution: 14s Result series: 1
Evaluation time
Remove Panel
Execute
Table
Graph
Load time: 44ms Resolution: 14s Result series: 1
Evaluation time
Remove Panel

```
kubeconeu2022_demo{type="med_res"} @ 1652792520
```

2022-05-17 12:35:00

```
kubeconeu2022_demo{instance="localhost:8080", job="sparse_histograms", type="med_res"}    { count:8 sum:84.4 (0.5,1]:2 (2,4]:3 (16,32]:3 }
```

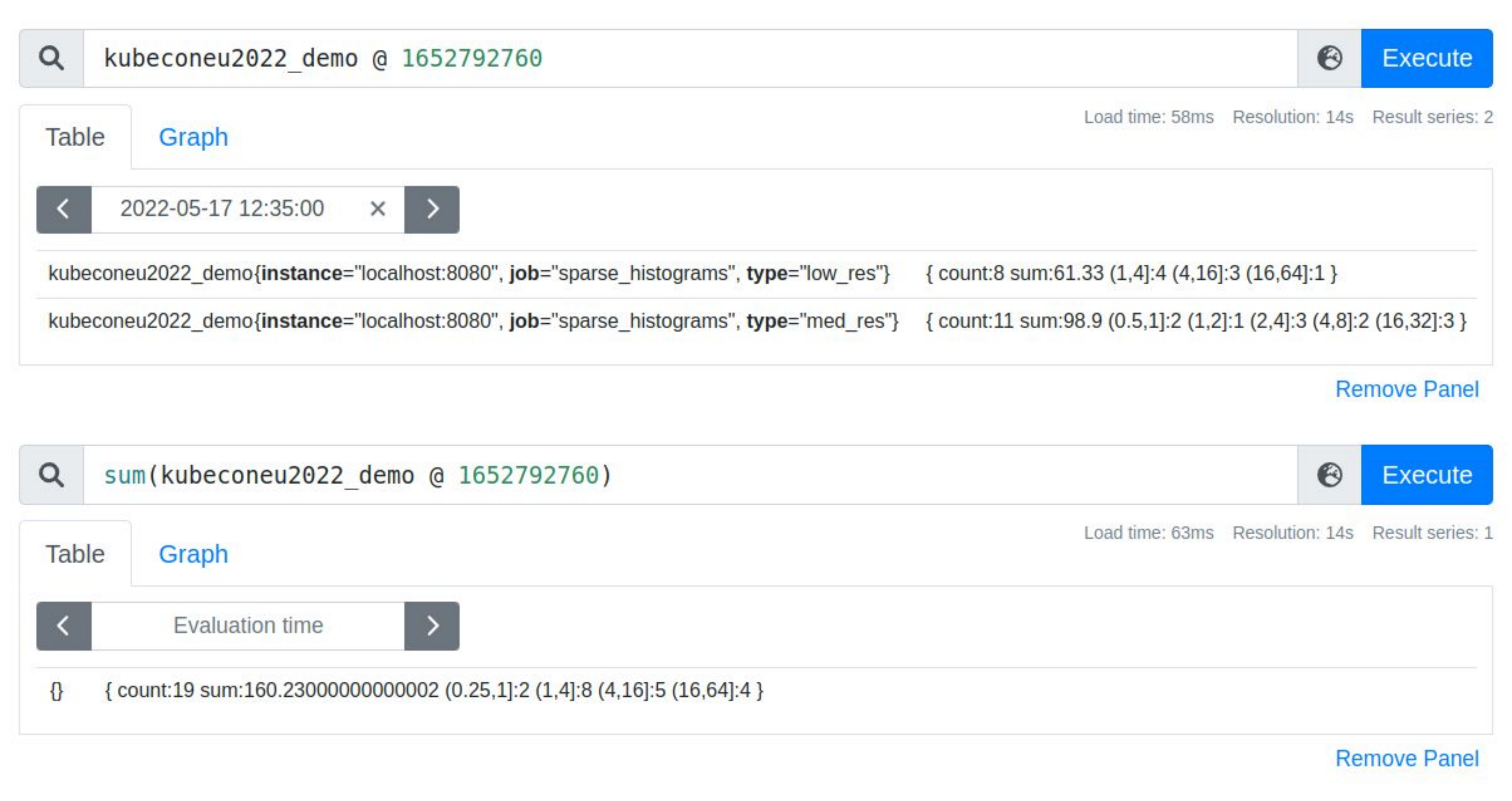
```
kubeconeu2022_demo{type="med_res"} @ 1652792640
```

```
kubeconeu2022_demo{instance="localhost:8080", job="sparse_histograms", type="med_res"}    { count:6 sum:89.5 (1,2]:1 (4,8]:2 (16,32]:3 }
```

```
kubeconeu2022_demo{type="med_res"} @ 1652792520 +
kubeconeu2022_demo{type="med_res"} @ 1652792640
```

```
{instance="localhost:8080", job="sparse_histograms", type="med_res"}    { count:14 sum:173.9 (0.5,1]:2 (1,2]:1 (2,4]:3 (4,8]:2 (16,32]:6 }
```

@_codesome



```
kubeconeu2022_demo @ 1652792760
```

Load time: 58ms Resolution: 14s Result series: 2

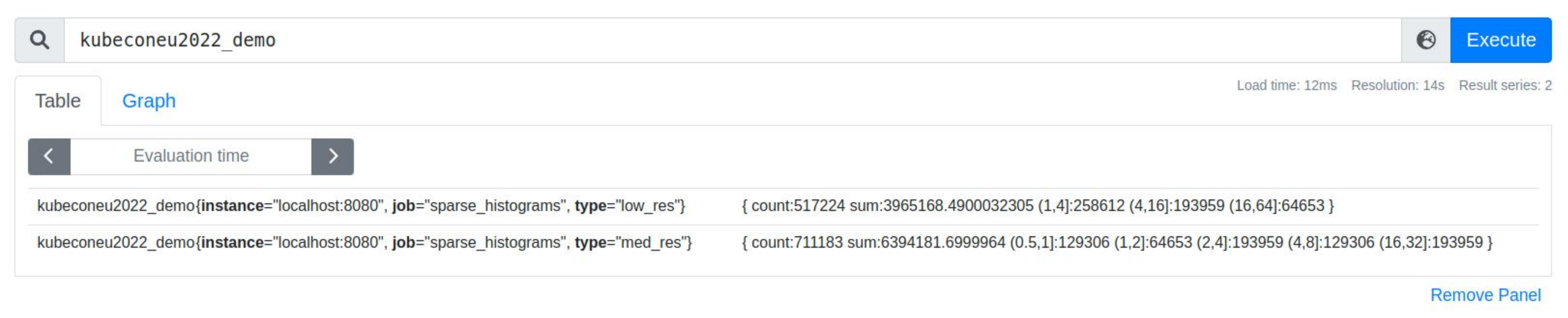
2022-05-17 12:35:00

| | |
|---|---|
| kubeconeu2022_demo{**instance**="localhost:8080", **job**="sparse_histograms", **type**="low_res"} | { count:8 sum:61.33 (1,4]:4 (4,16]:3 (16,64]:1 } |
| kubeconeu2022_demo{**instance**="localhost:8080", **job**="sparse_histograms", **type**="med_res"} | { count:11 sum:98.9 (0.5,1]:2 (1,2]:1 (2,4]:3 (4,8]:2 (16,32]:3 } |

```
sum(kubeconeu2022_demo @ 1652792760)
```

Load time: 63ms Resolution: 14s Result series: 1

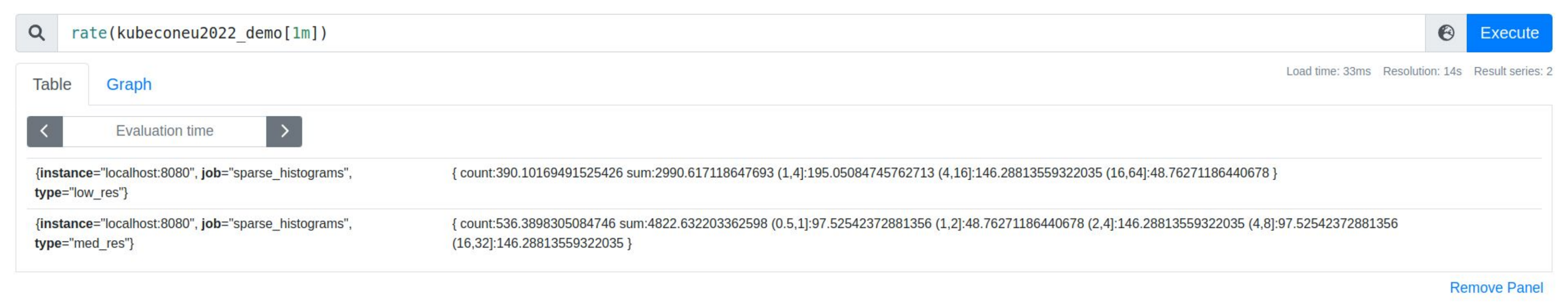
| | |
|---|---|
| {} | { count:19 sum:160.23000000000002 (0.25,1]:2 (1,4]:8 (4,16]:5 (16,64]:4 } |



```
kubeconeu2022_demo
```

Load time: 12ms  Resolution: 14s  Result series: 2

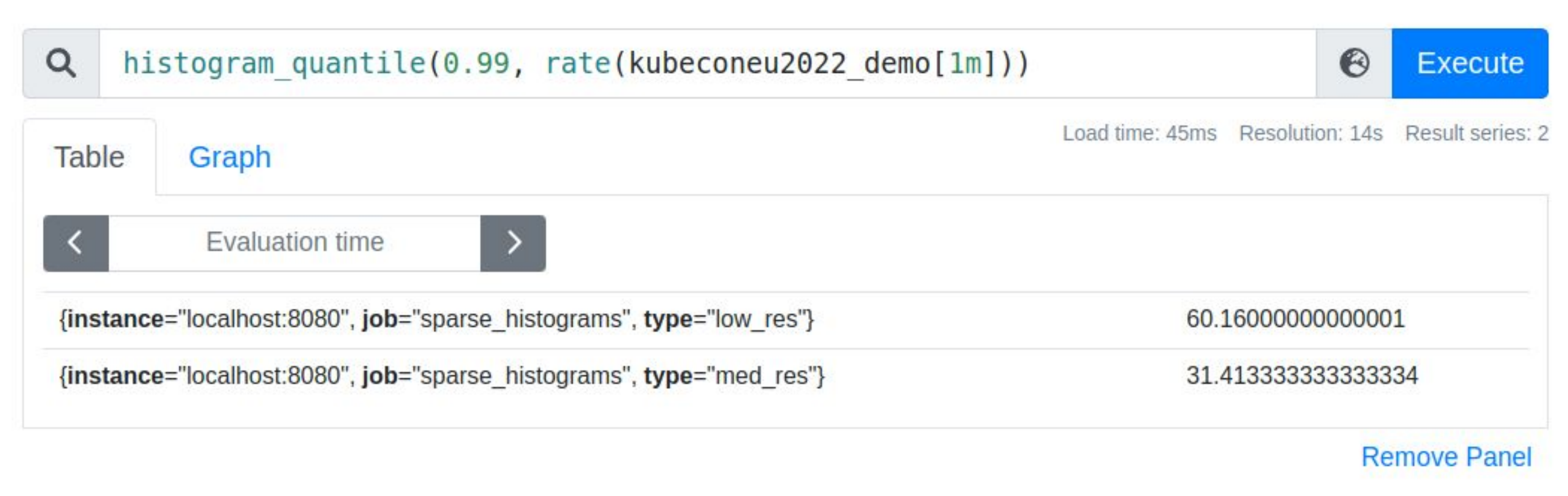
| | |
|---|---|
| kubeconeu2022_demo{**instance**="localhost:8080", **job**="sparse_histograms", **type**="low_res"} | { count:517224 sum:3965168.4900032305 (1,4]:258612 (4,16]:193959 (16,64]:64653 } |
| kubeconeu2022_demo{**instance**="localhost:8080", **job**="sparse_histograms", **type**="med_res"} | { count:711183 sum:6394181.6999964 (0.5,1]:129306 (1,2]:64653 (2,4]:193959 (4,8]:129306 (16,32]:193959 } |



```
rate(kubeconeu2022_demo[1m])
```

Load time: 33ms  Resolution: 14s  Result series: 2

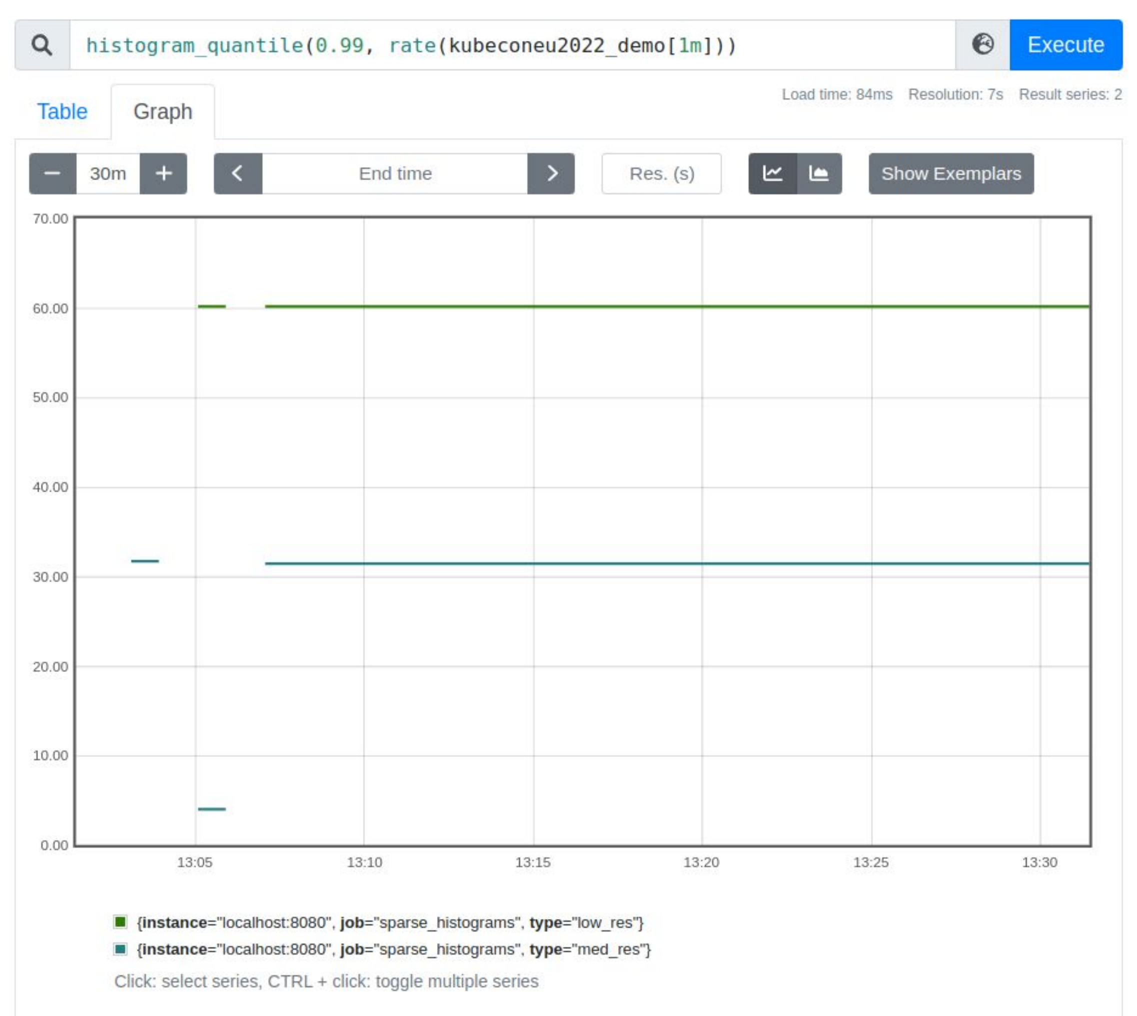
| | |
|---|---|
| {**instance**="localhost:8080", **job**="sparse_histograms", **type**="low_res"} | { count:390.10169491525426 sum:2990.617118647693 (1,4]:195.05084745762713 (4,16]:146.28813559322035 (16,64]:48.7627118644O678 } |
| {**instance**="localhost:8080", **job**="sparse_histograms", **type**="med_res"} | { count:536.3898305084746 sum:4822.632203362598 (0.5,1]:97.52542372881356 (1,2]:48.76271186440678 (2,4]:146.28813559322035 (4,8]:97.52542372881356 (16,32]:146.28813559322035 } |



```
histogram_quantile(0.99, rate(kubeconeu2022_demo[1m]))
```

Load time: 45ms   Resolution: 14s   Result series: 2

| | |
|---|---|
| {**instance**="localhost:8080", **job**="sparse_histograms", **type**="low_res"} | 60.16000000000001 |
| {**instance**="localhost:8080", **job**="sparse_histograms", **type**="med_res"} | 31.413333333333334 |



```
histogram_quantile(0.99, rate(kubeconeu2022_demo[1m]))
```

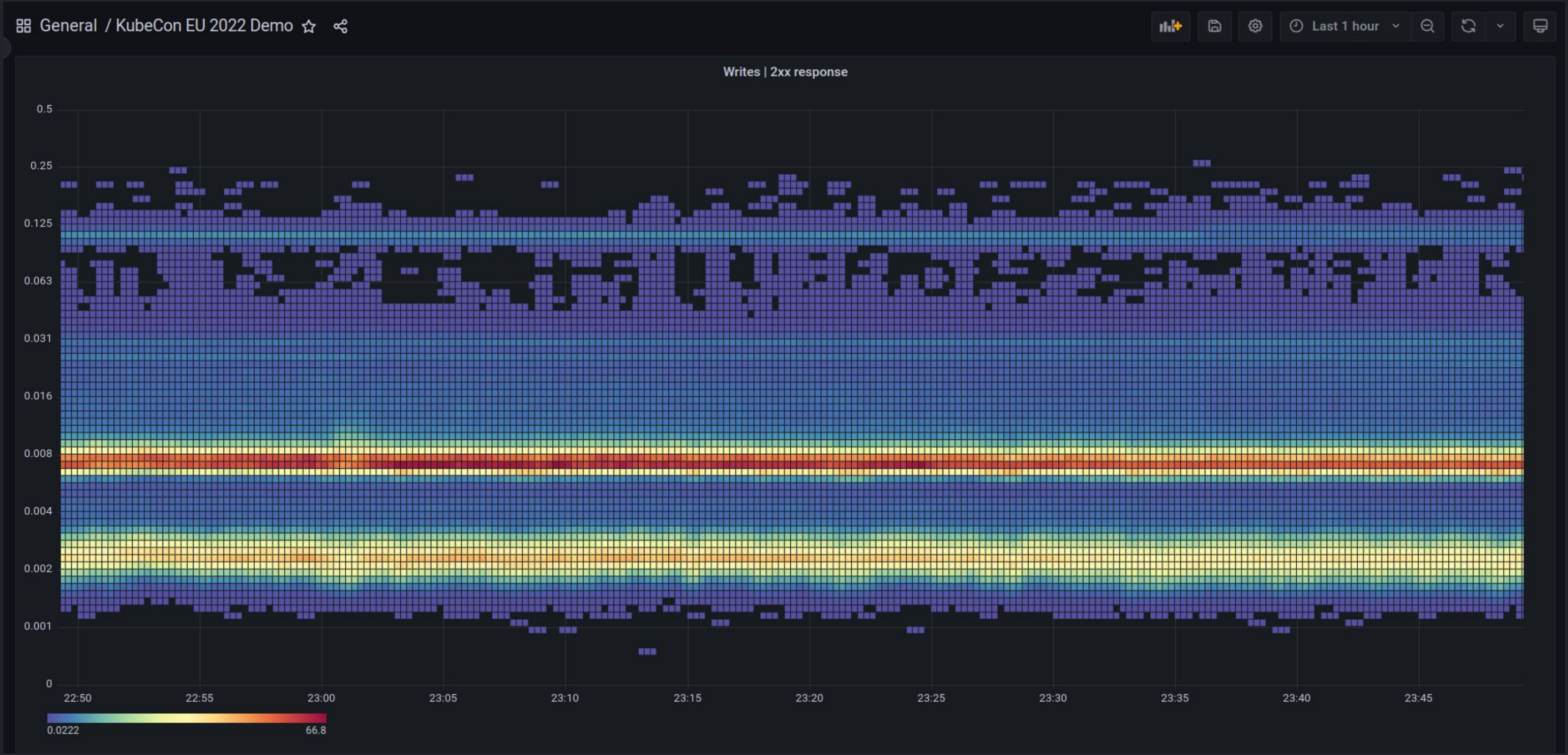
Load time: 84ms   Resolution: 7s   Result series: 2

🟩 {**instance**="localhost:8080", **job**="sparse_histograms", **type**="low_res"}

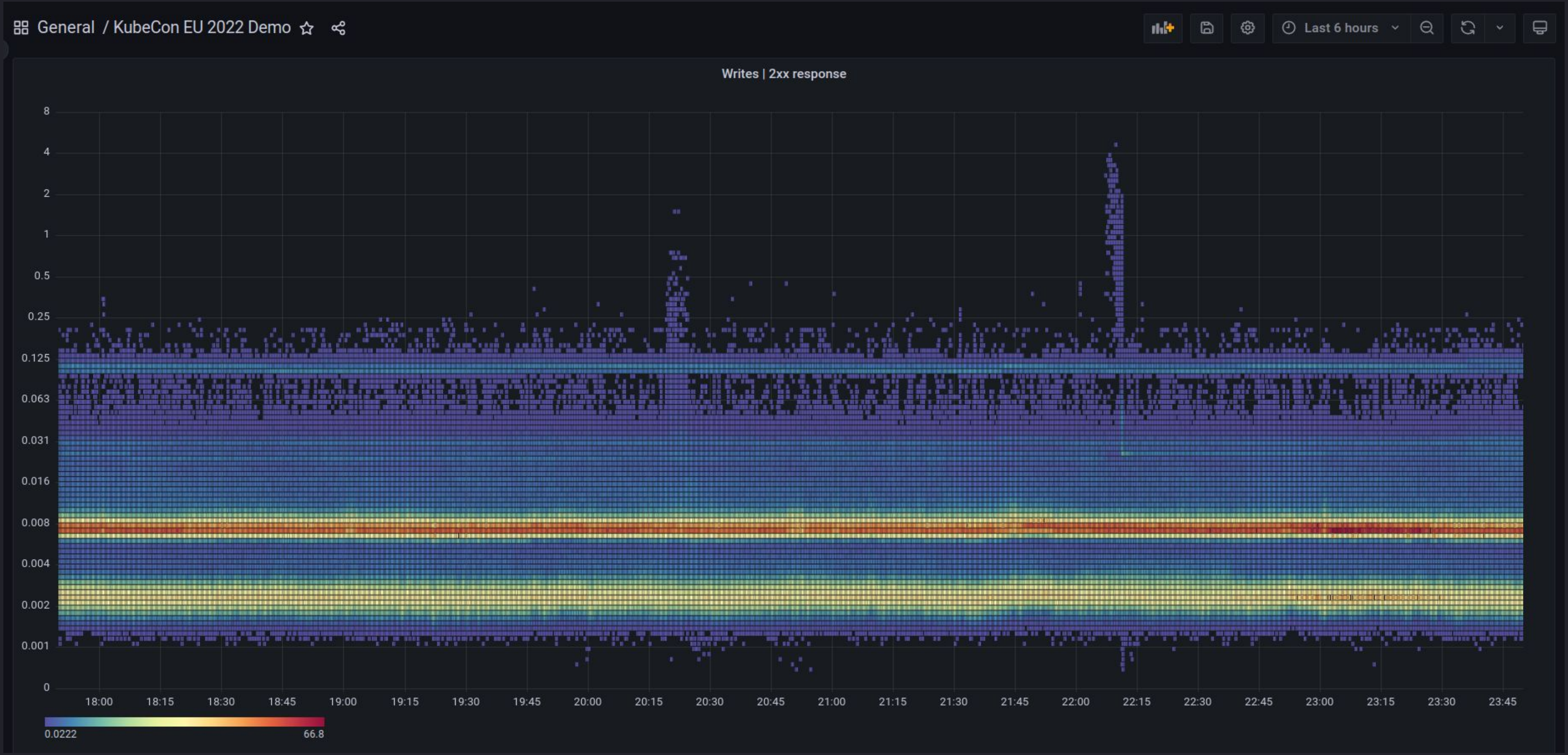
🟩 {**instance**="localhost:8080", **job**="sparse_histograms", **type**="med_res"}

Click: select series, CTRL + click: toggle multiple series

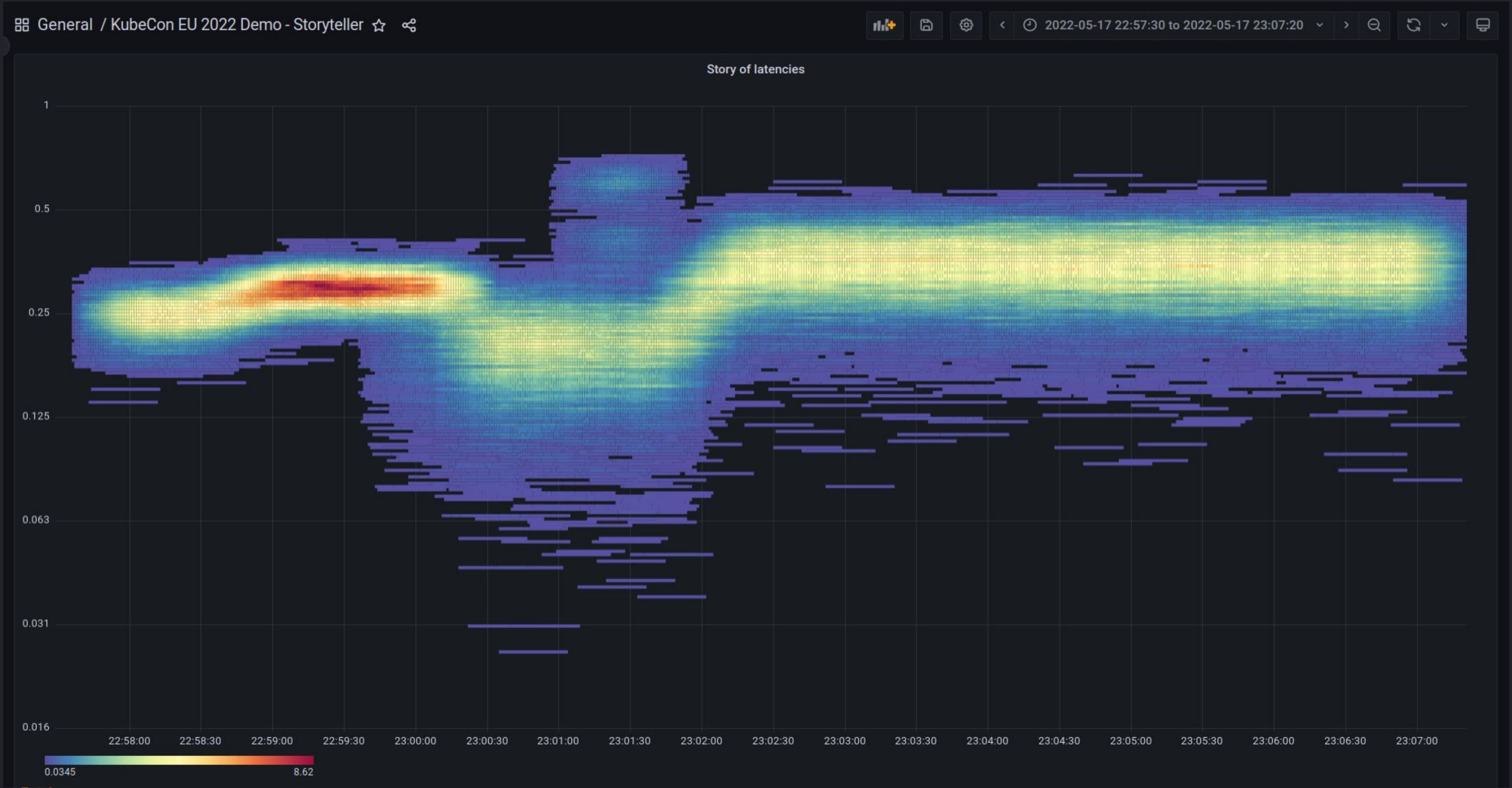
@_codesome


General / KubeCon EU 2022 Demo
Last 1 hour
Writes | 2xx response
0.5
0.25
0.125
0.063
0.031
0.016
0.008
0.004
0.002
0.001
0
22:50
22:55
23:00
23:05
23:10
23:15
23:20
23:25
23:30
23:35
23:40
23:45
0.0222
66.8

@_codesome


General / KubeCon EU 2022 Demo
Last 6 hours
Writes | 2xx response
8
4
2
1
0.5
0.25
0.125
0.063
0.031
0.016
0.008
0.004
0.002
0.001
0
18:00
18:15
18:30
18:45
19:00
19:15
19:30
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
21:45
22:00
22:15
22:30
22:45
23:00
23:15
23:30
23:45
0.0222
66.8

@_codesome



General / KubeCon EU 2022 Demo - Storyteller

2022-05-17 22:57:30 to 2022-05-17 23:07:20

@_codesome

# How can I use it?

It is still in a Proof of Concept stage (but likely gonna stick).

Available here:

Instrumentation: https://github.com/prometheus/client_golang/tree/sparsehistogram

Prometheus server: https://github.com/prometheus/prometheus/tree/sparsehistogram

New heatmaps: main branch of grafana/grafana

Grafana Labs

# THANK YOU! Questions?