The Sixth Sense: Finding “Dead” Metrics and Silent Alerts in Your Monitoring

Roman Khavronenko, Engineering Manager

VICTORIA METRICS

The High Performance
Open Source Time Series Database & Monitoring Solution

20k stars
1000+ contributors

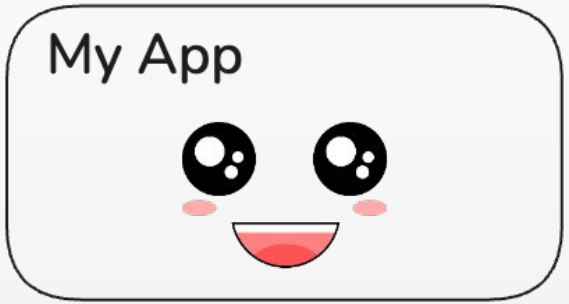
100M+ downloads

Grammarly, Roblox,
Wix, Hubspot, Semrush

Observability is expensive

Observability is expensive

Engineer runs his own app



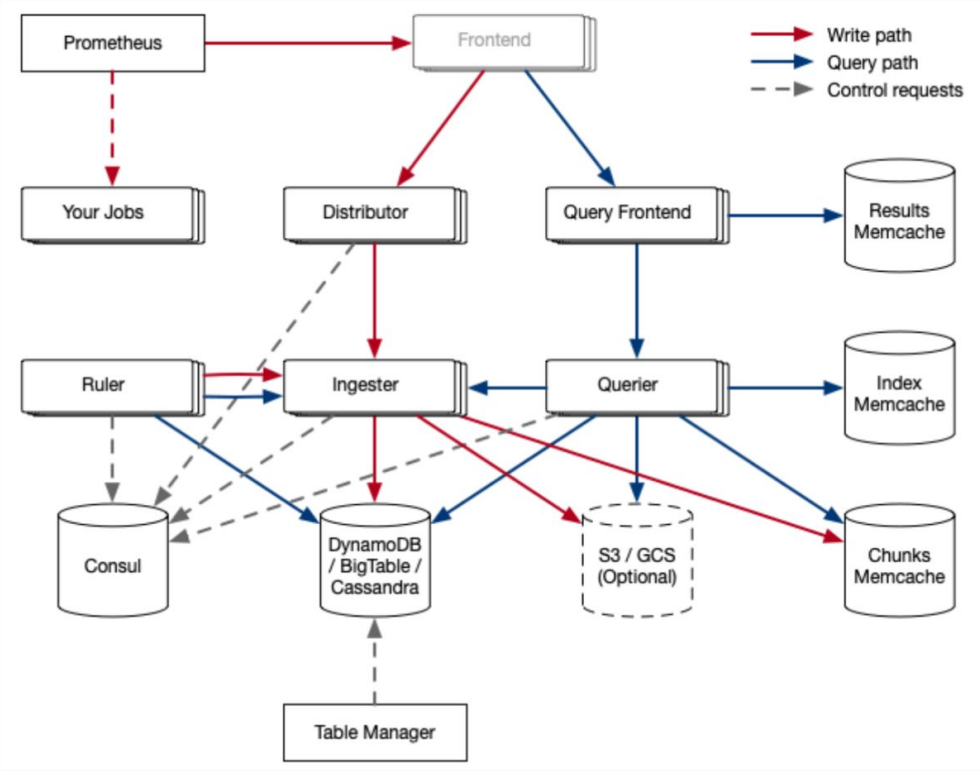
- Are you okay?

```
tail -f logs.txt
```

Why observability became expensive

- Industry is driven by Big Tech companies
- Tech giants were built in era of cheap money, concerned by **scale**, **not efficiency**
- Tech giants promote **complex systems**, designed for infinite scalability, encouraging **waste of resources**.

Engineer runs "helm install"



- What are you???

Why observability is expensive

- Systems become more and **more complex**
- More complex systems, **more telemetry is needed**
- More telemetry, **more complex monitoring systems**
- Systems are built out of **"blocks"**
- **"Blocks"** are tailored for many use cases, are more complex and less efficient
- Telemetry is emitted **just in case**, because **no one knows what** and **when** they'd need
- Every emitted metric **contributes to monthly costs** and **to cognitive load.**

Observability vendors are focused on symptoms

- More signals to collect and process
- More efficient databases engines, to deal with more data
- Better compression, to store more data
- Infinite scalability for infinite cost

O11y spend should be 20-30% of infra spend https://www.honeycomb.io/blog/how-much-should-my-observability-stack-cost

What is the current trend for simplifying observability?

- Use AI…
- Maybe AI can make writing queries easier?
- Maybe AI can read all the metrics we have and answer questions?
- Maybe AI can **understand how it works?**

What is the current trend for simplifying observability?

- Use AI…
- Maybe AI can make writing queries easier?
- Maybe AI can read all the metrics we have and answer questions?
- Maybe AI can **understand how it works?**

Complexity creates more complexity

The average observability experience

How is my new k8s cluster?

```
kubectl get po -wA
```

The average observability experience

```
NAMESPACE          NAME                                                      READY   STATUS
gke-managed-cim    kube-state-metrics-0                                      2/2     Running
gmp-system         collector-wrp5g                                           2/2     Running
gmp-system         collector-zf7vw                                           2/2     Running
gmp-system         gmp-operator-84445969dc-ndk4r                             1/1     Running
kube-system        container-watcher-hhm7p                                   1/1     Running
kube-system        container-watcher-qj89f                                   1/1     Running
kube-system        event-exporter-gke-6999444f65-7hq5h                       2/2     Running
kube-system        fluentbit-gke-mnc9p                                       3/3     Running
kube-system        fluentbit-gke-rmkcj                                       3/3     Running
kube-system        gke-metrics-agent-mtxll                                   3/3     Running
kube-system        gke-metrics-agent-qj5kv                                   3/3     Running
kube-system        konnectivity-agent-65699745c-7mrff                        2/2     Running
kube-system        konnectivity-agent-65699745c-w7dhp                        2/2     Running
kube-system        konnectivity-agent-autoscaler-8597594bf4-8hmw2            1/1     Running
kube-system        kube-dns-7b6d76f989-n9rpz                                 4/4     Running
kube-system        kube-dns-7b6d76f989-vc4wb                                 4/4     Running
kube-system        kube-dns-autoscaler-58958dfdbd-wvxjb                      1/1     Running
kube-system        kube-proxy-gke-osod-demo-osod-demo-pool-55cdde72-023q      1/1     Running
kube-system        kube-proxy-gke-osod-demo-osod-demo-pool-55cdde72-qn9f      1/1     Running
kube-system        l7-default-backend-7b68f77db9-ljc6r                       1/1     Running
kube-system        metrics-server-v1.33.0-57475bcb6d-rbvcf                   1/1     Running
kube-system        pdcsi-node-cqp6p                                          2/2     Running
kube-system        pdcsi-node-zmghd                                          2/2     Running
```

The average observability experience

How to monitor k8s?

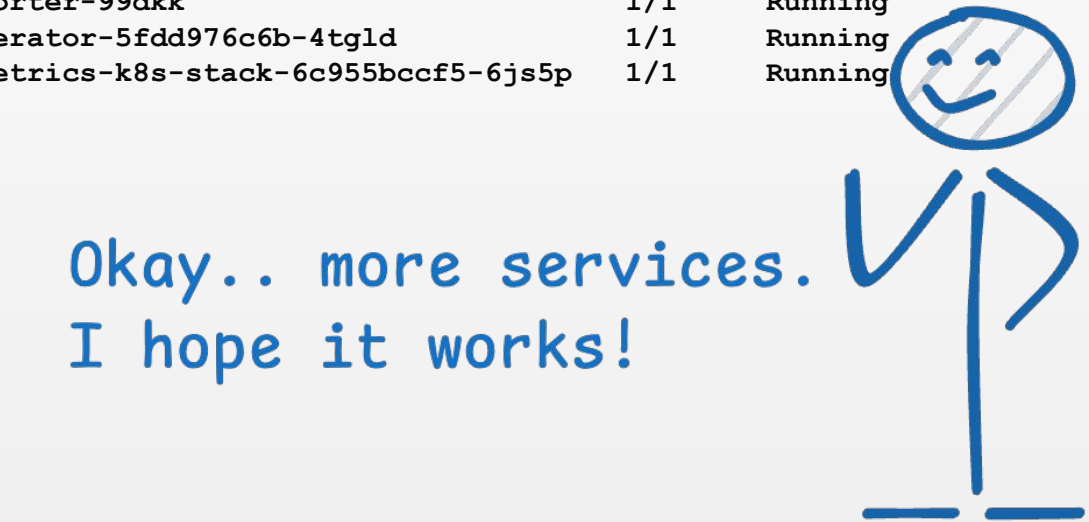
Helm Charts / VictoriaMetrics K8s Stack

This chart is an All-in-one solution to start monitoring kubernetes cluster. It installs multiple dependency charts like grafana , node-exporter , kube-state-metrics and victoria-metrics-operator .

Okay... helm install

The average observability experience

```
NAMESPACE        NAME                                                     READY   STATUS
k8s-stack-demo   vmagent-vmks-victoria-metrics-k8s-stack-b5ffdb7cc-27rwr   2/2     Running
k8s-stack-demo   vmalert-vmks-victoria-metrics-k8s-stack-59585875bd-pxrmk  2/2     Running
k8s-stack-demo   vmalertmanager-vmks-victoria-metrics-k8s-stack-0         2/2     Running
k8s-stack-demo   vmks-grafana-774596749-rpvsw                             2/2     Running
k8s-stack-demo   vmks-kube-state-metrics-748655b65b-ptbrx                 1/1     Running
k8s-stack-demo   vmks-prometheus-node-exporter-5dprv                      1/1     Running
k8s-stack-demo   vmks-prometheus-node-exporter-99dkk                      1/1     Running
k8s-stack-demo   vmks-victoria-metrics-operator-5fdd976c6b-4tgld          1/1     Running
k8s-stack-demo   vmsingle-vmks-victoria-metrics-k8s-stack-6c955bccf5-6js5p 1/1     Running
```

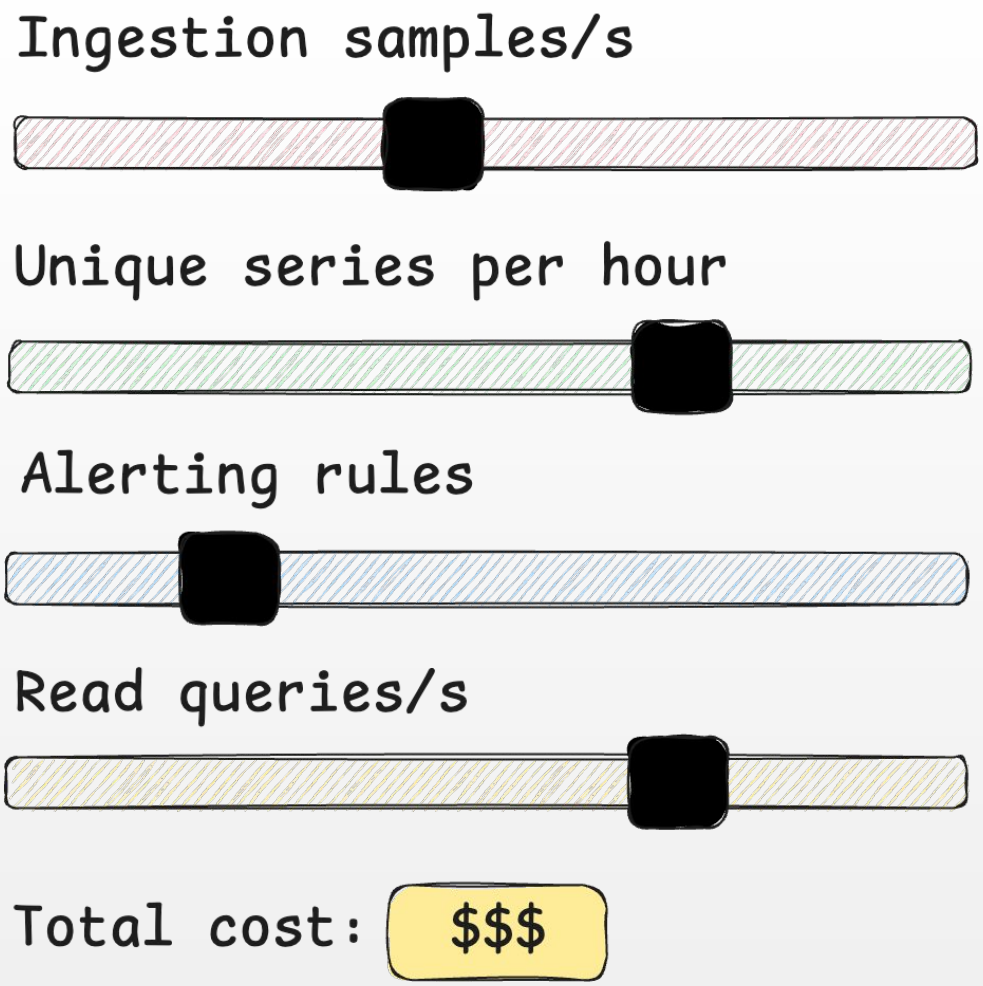


No such thing as a free lunch

More data means higher expenses

Every sample you collect increases expenses

It will be **sad to pay** for something **you don't ever use**.



How metrics are used?

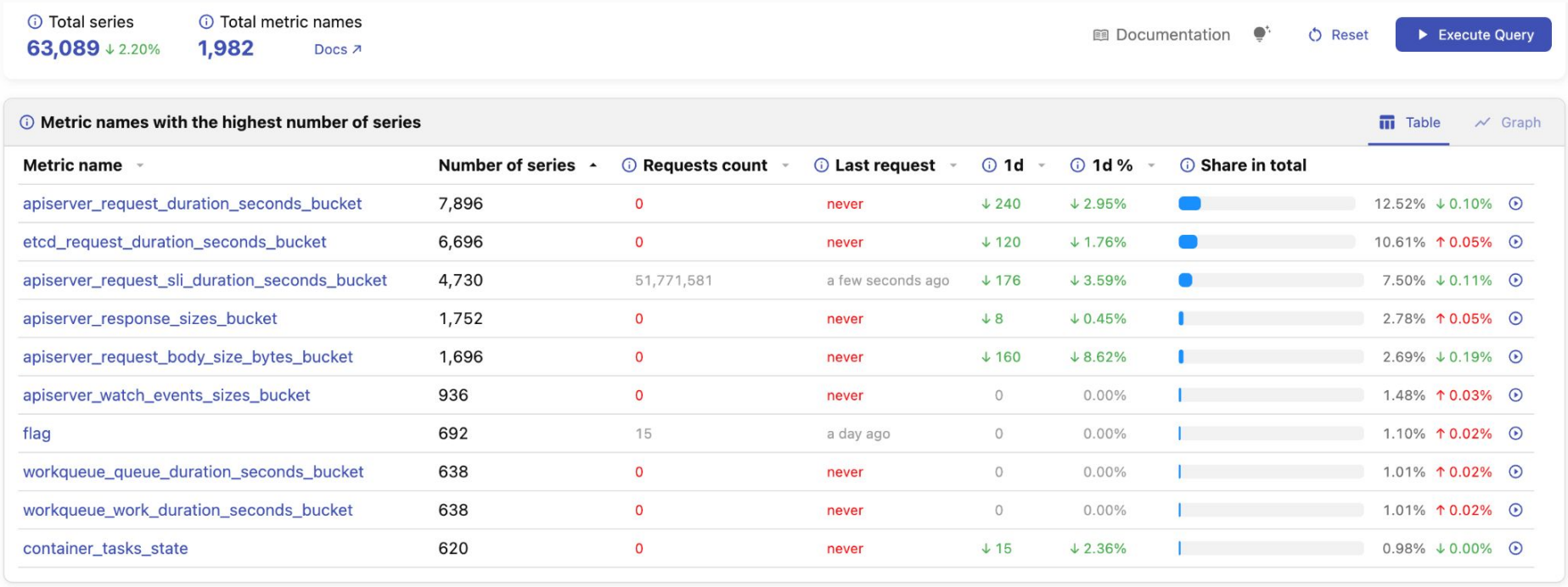
- In **alerting rules**, to ping you if something is wrong
- In **recording rules**, to speed up alerting rules or dashboards
- On **dashboards**, for system overview
- In **ad-hoc queries**, during investigation or incidents

Unused metric is a metric that was never queried

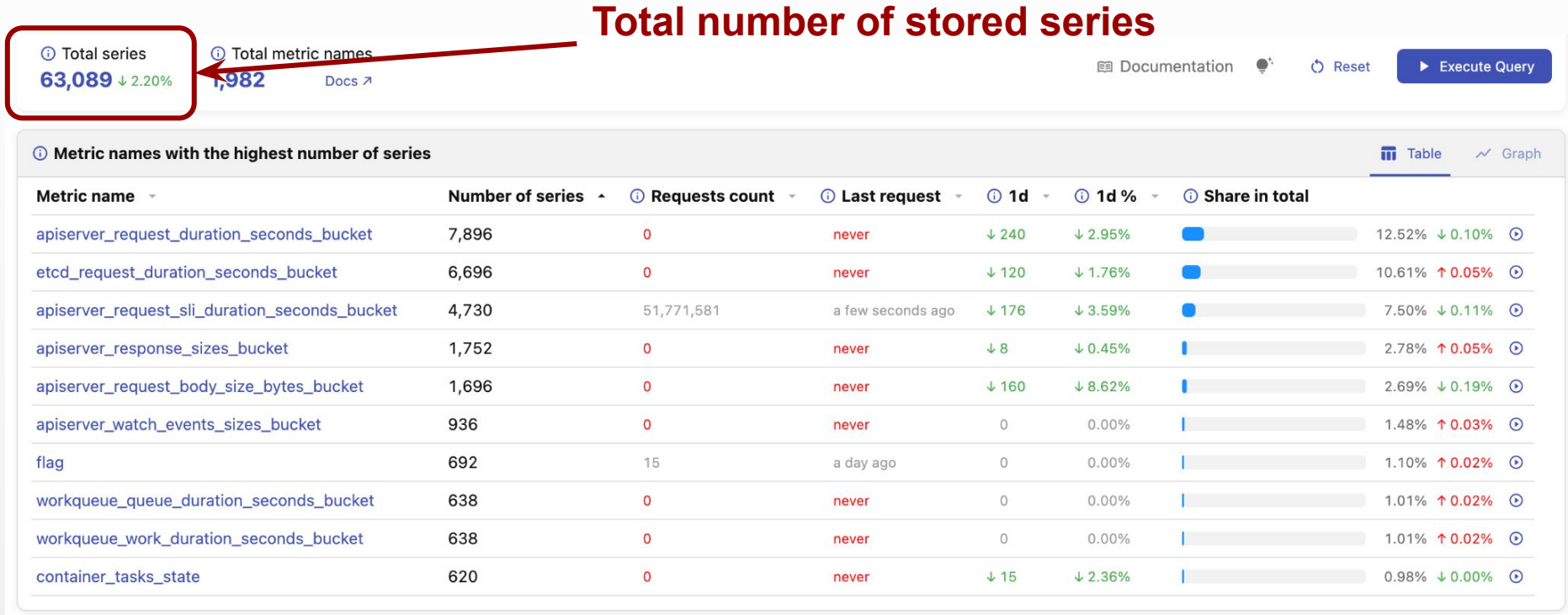
How to find unused metrics?

- Scan **Rules** and **dashboards** for queries the use
  - Build a whitelist of metric names
- **Ad-hoc queries** - ?

VictoriaMetrics tracks metric usage out of the box



Metric name	Number of series	Requests count	Last request	1d	1d %	Share in total
apiserver_request_duration_seconds_bucket	7,896	0	never	↓ 240	↓ 2.95%	12.52% ↓ 0.10%
etcd_request_duration_seconds_bucket	6,696	0	never	↓ 120	↓ 1.76%	10.61% ↑ 0.05%
apiserver_request_sli_duration_seconds_bucket	4,730	51,771,581	a few seconds ago	↓ 176	↓ 3.59%	7.50% ↓ 0.11%
apiserver_response_sizes_bucket	1,752	0	never	↓ 8	↓ 0.45%	2.78% ↑ 0.05%
apiserver_request_body_size_bytes_bucket	1,696	0	never	↓ 160	↓ 8.62%	2.69% ↓ 0.19%
apiserver_watch_events_sizes_bucket	936	0	never	0	0.00%	1.48% ↑ 0.03%
flag	692	15	a day ago	0	0.00%	1.10% ↑ 0.02%
workqueue_queue_duration_seconds_bucket	638	0	never	0	0.00%	1.01% ↑ 0.02%
workqueue_work_duration_seconds_bucket	638	0	never	0	0.00%	1.01% ↑ 0.02%
container_tasks_state	620	0	never	↓ 15	↓ 2.36%	0.98% ↓ 0.00%


Total number of stored series
Total series
63,089 ↓ 2.20%
Total metric names
1,982
Docs ↗
Documentation
Reset
Execute Query
Metric names with the highest number of series
Table
Graph
Metric name
Number of series
Requests count
Last request
1d
1d %
Share in total
apiserver_request_duration_seconds_bucket 7,896 0 never ↓ 240 ↓ 2.95% 12.52% ↓ 0.10%
etcd_request_duration_seconds_bucket 6,696 0 never ↓ 120 ↓ 1.76% 10.61% ↑ 0.05%
apiserver_request_sli_duration_seconds_bucket 4,730 51,771,581 a few seconds ago ↓ 176 ↓ 3.59% 7.50% ↓ 0.11%
apiserver_response_sizes_bucket 1,752 0 never ↓ 8 ↓ 0.45% 2.78% ↑ 0.05%
apiserver_request_body_size_bytes_bucket 1,696 0 never ↓ 160 ↓ 8.62% 2.69% ↓ 0.19%
apiserver_watch_events_sizes_bucket 936 0 never 0 0.00% 1.48% ↑ 0.03%
flag 692 15 a day ago 0 0.00% 1.10% ↑ 0.02%
workqueue_queue_duration_seconds_bucket 638 0 never 0 0.00% 1.01% ↑ 0.02%
workqueue_work_duration_seconds_bucket 638 0 never 0 0.00% 1.01% ↑ 0.02%
container_tasks_state 620 0 never ↓ 15 ↓ 2.36% 0.98% ↓ 0.00%

Top N metric names by number of series

Total series
63,089 ↓ 2.20%

Total metric names
1,982
Docs ↗

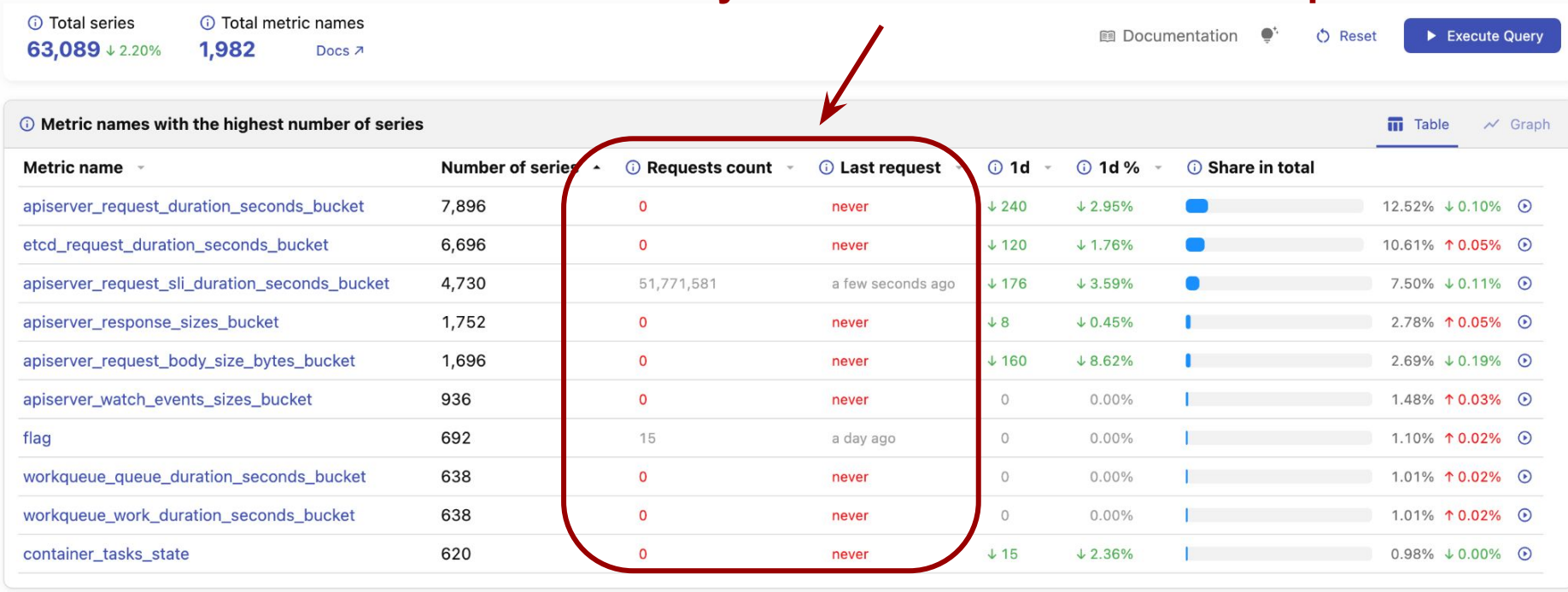
Documentation | Reset | Execute Query

Metric names with the highest number of series

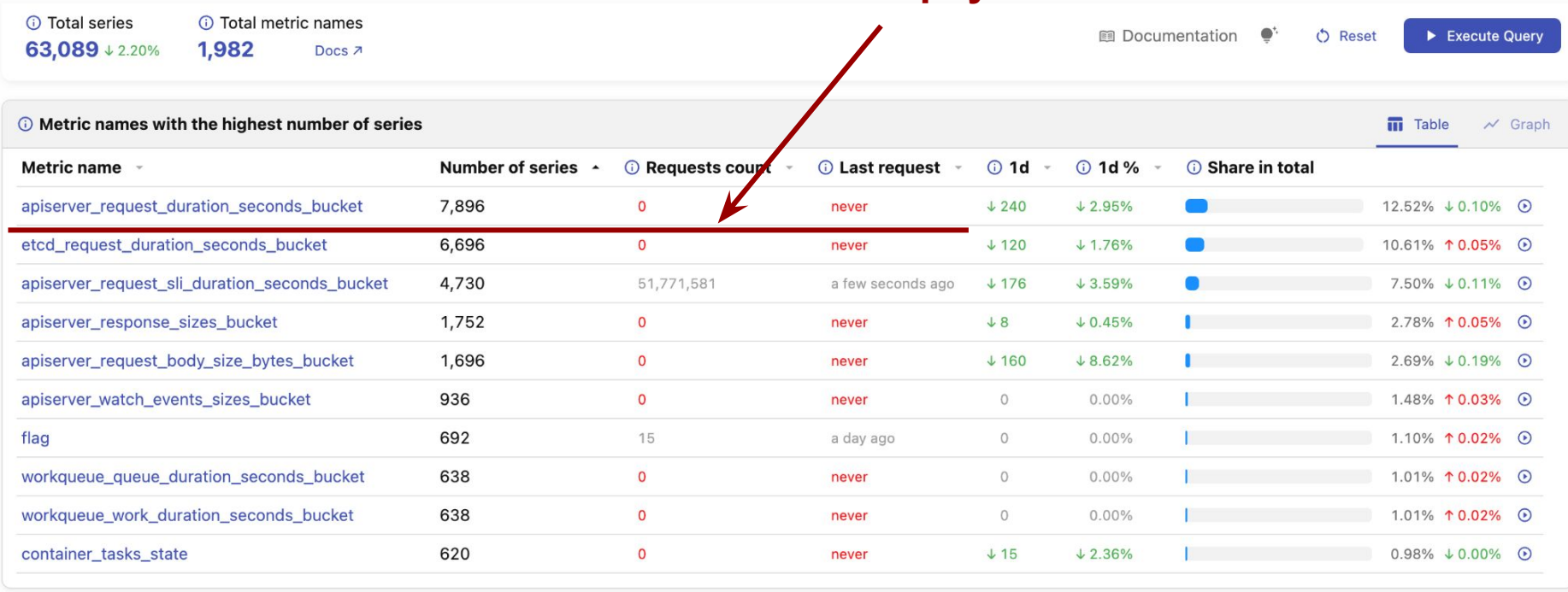
Table | Graph

Metric name	Number of series	Requests count	Last request	1d	1d %	Share in total
apiserver_request_duration_seconds_bucket	7,896	0	never	↓ 240	↓ 2.95%	12.52% ↓ 0.10%
etcd_request_duration_seconds_bucket	6,696	0	never	↓ 120	↓ 1.76%	10.61% ↑ 0.05%
apiserver_request_sli_duration_seconds_bucket	4,730	1,771,581	a few seconds ago	↓ 176	↓ 3.59%	7.50% ↓ 0.11%
apiserver_response_sizes_bucket	1,752	0	never	↓ 8	↓ 0.45%	2.78% ↑ 0.05%
apiserver_request_body_size_bytes_bucket	1,696	0	never	↓ 160	↓ 8.62%	2.69% ↓ 0.19%
apiserver_watch_events_sizes_bucket	936	0	never	0	0.00%	1.48% ↑ 0.03%
flag	692	15	a day ago	0	0.00%	1.10% ↑ 0.02%
workqueue_queue_duration_seconds_bucket	638	0	never	0	0.00%	1.01% ↑ 0.02%
workqueue_work_duration_seconds_bucket	638	0	never	0	0.00%	1.01% ↑ 0.02%
container_tasks_state	620	0	never	↓ 15	↓ 2.36%	0.98% ↓ 0.00%

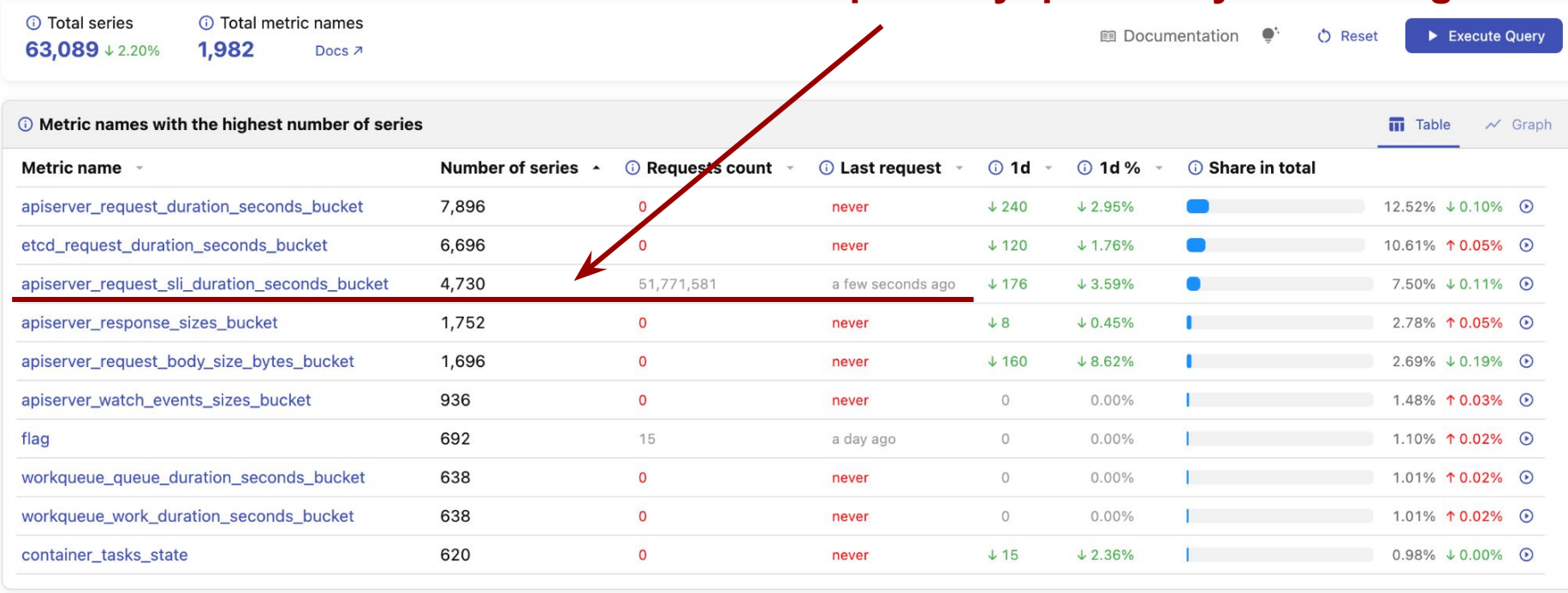
How many times and when metric was queried



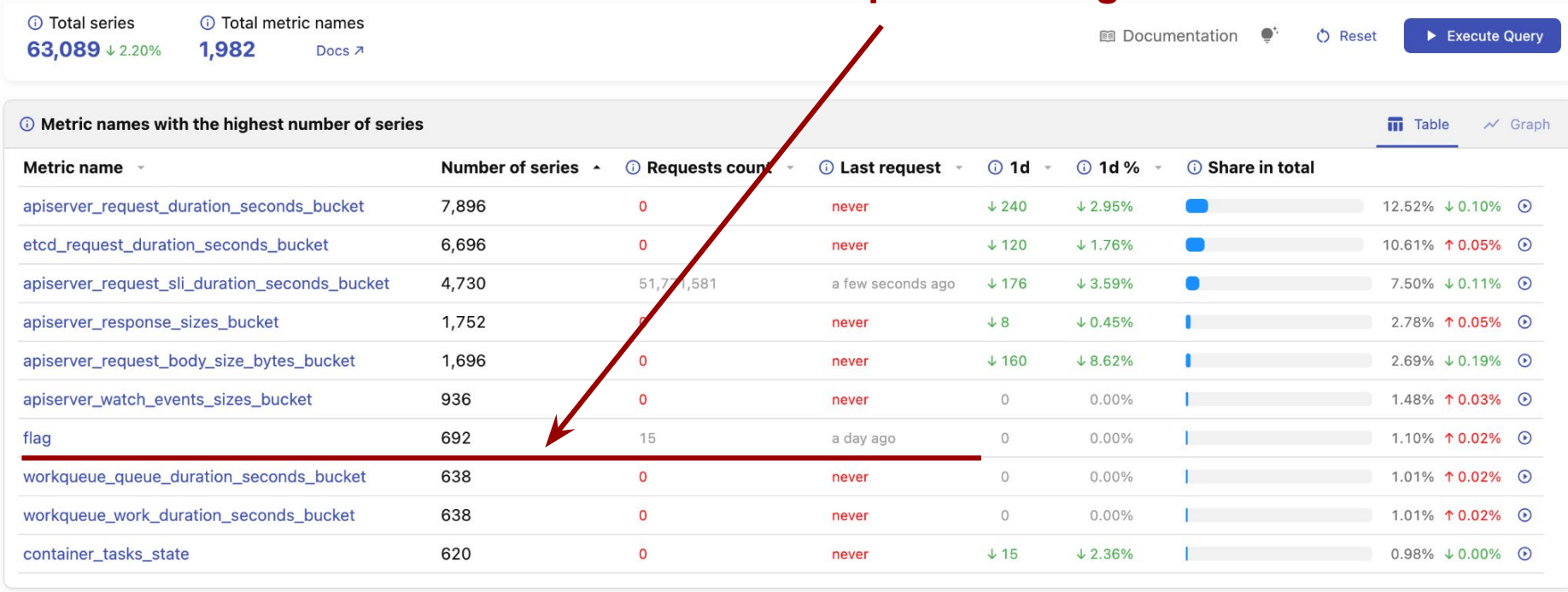
This metric doesn't pay its rent!



This metric is repeatedly queried by something



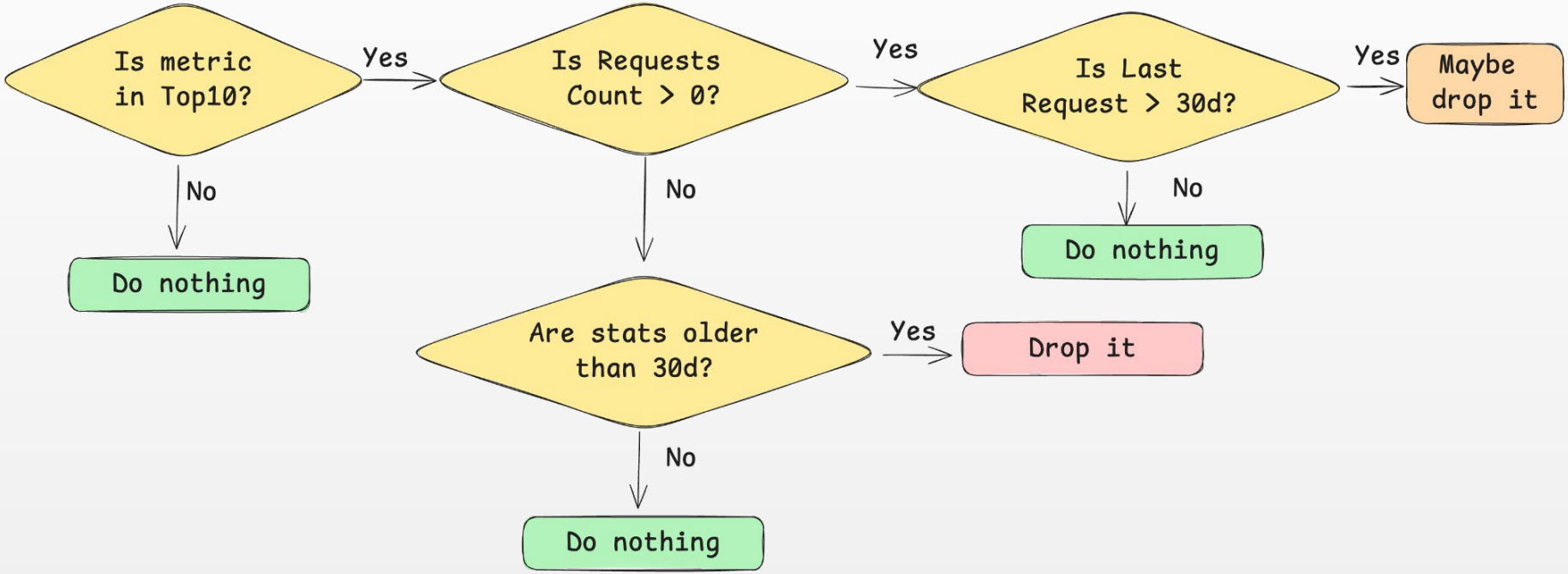
This metric was queried 1d ago



Demo time!

http://localhost:8428/vmui

How to use Cardinality Explorer



VictoriaMetrics "Cardinality explorer"

- Built-in view in VictoriaMetrics UI
- Enabled by default
- Shows amount of metrics and time series you store
- Shows **when** and **how many times** metric was queried
- Allows finding metrics that are:
  - **never** queried
  - **occasionally** queried
  - **repeatedly** queried
- Metric name usage is exposed via API and can be used programmatically

Alerting

A typical alerting rule

```
- alert: KubeControlManagerIsDown
  expr: up{job="kube-controler-manager"} == 0
```

This alert will never fire… But why?

A typical alerting rule

```
- alert: KubeControlManagerIsDown
  expr: up{job="kube-controler-manager"} == 0
```

This alert will never fire… But why?

1. Controller manager will never go down!

A typical alerting rule

```
- alert: KubeControlManagerIsDown
  expr: up{job="kube-controler-manager"} == 0
```

This alert will never fire… But why?

1. Controller manager will never go down!
2. We don't scrape metrics for Controller manager

A typical alerting rule

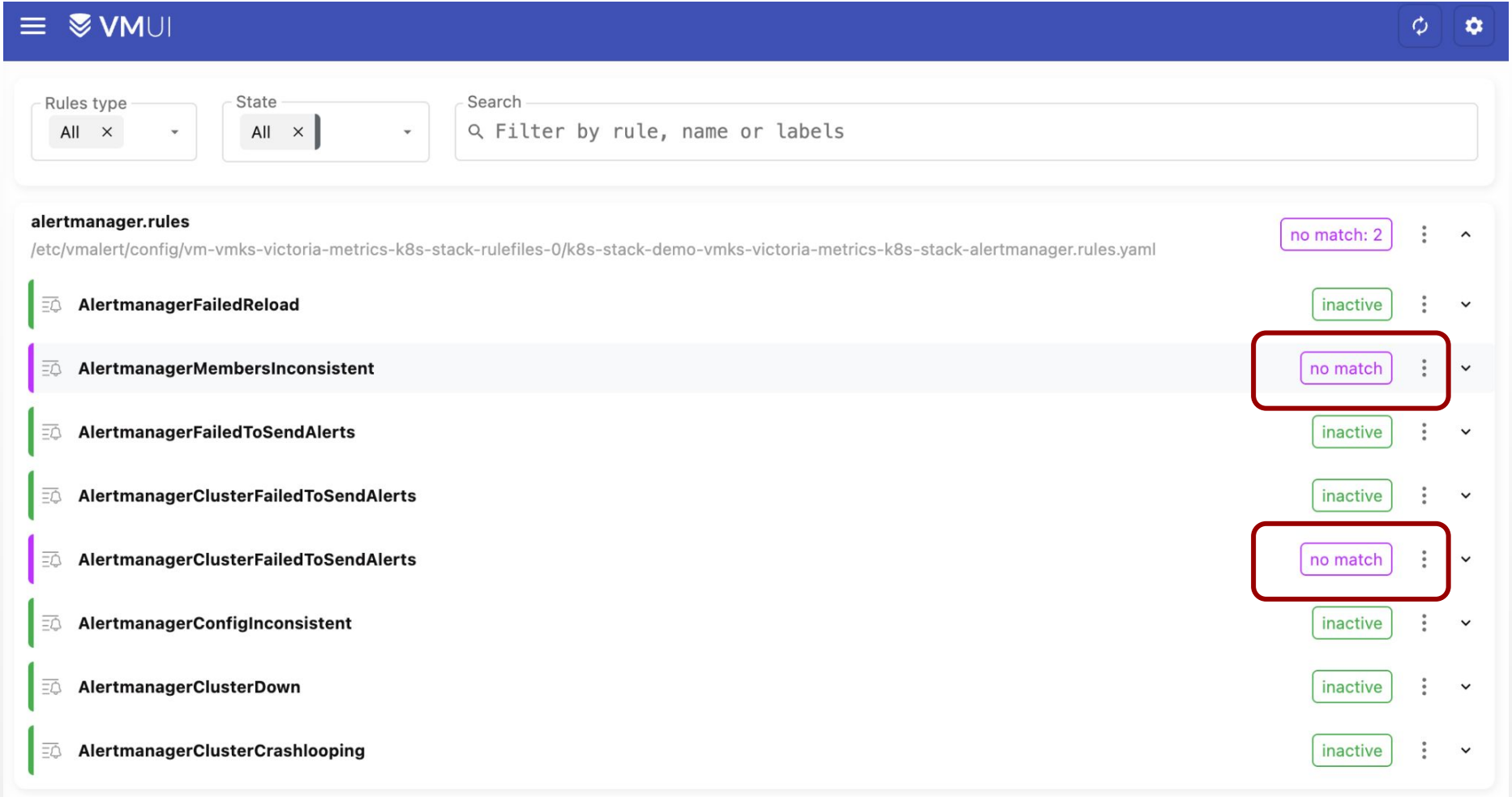
```
- alert: KubeControlManagerIsDown

  expr: up{job="kube-controler-manager"} == 0
```

This alert will never fire… But why?

1. Controller manager will never go down!
2. We don't scrape metrics for Controller manager
3. Job name should have two `L` letters:
   `up{job="kube-controller-manager"}`

VictoriaMetrics alerting page

Group has 2 rules that don't match any series at all


VMUI
Rules type
All
State
All
Search
Filter by rule, name or labels
alertmanager.rules
/etc/vmalert/config/vm-vmks-victoria-metrics-k8s-stack-rulefiles-0/k8s-stack-demo-vmks-victoria-metrics-k8s-stack-alertmanager.rules.yaml
no match: 2
AlertmanagerFailedReload
inactive
AlertmanagerMembersInconsistent
no match
AlertmanagerFailedToSendAlerts
inactive
AlertmanagerClusterFailedToSendAlerts
inactive
AlertmanagerClusterFailedToSendAlerts
no match
AlertmanagerConfigInconsistent
inactive
AlertmanagerClusterDown
inactive
AlertmanagerClusterCrashlooping
inactive

Demo time!

http://localhost:8428/vmui

VictoriaMetrics "Alerting" page

- Built-in view in VictoriaMetrics UI
- Shows all alerting and recording rules with their states
- Helps identifying alerting and recording rules that don't match any series

https://victoriametrics.com/blog/never-firing-alerts/

Never-firing alerts: What they are and how to deal with them

Detecting "bad" rules automatically

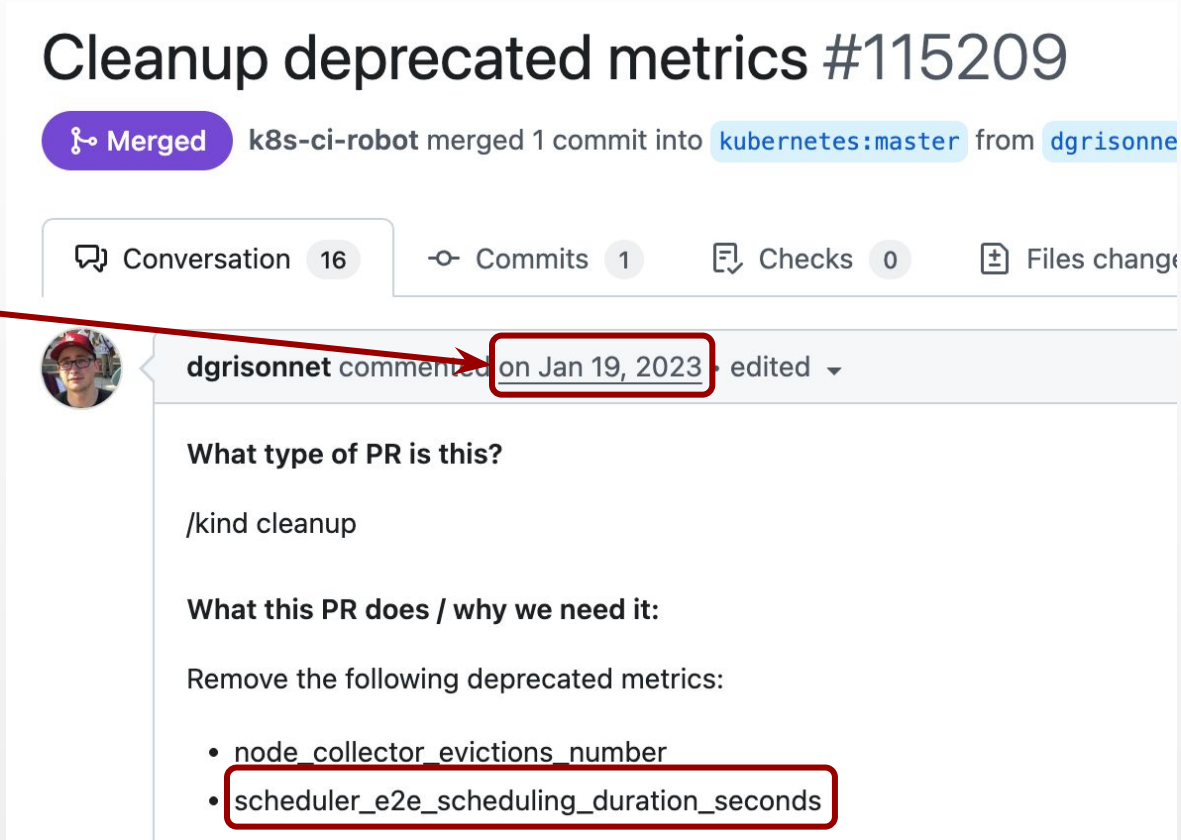
```
- alert: AlertMatchesNoSeries
  expr: max(vmalert_alerting_rules_last_evaluation_series_fetched)
        by(group, alertname) == 0

- alert: RecordingRuleProducesNoResults
  expr: sum(vmalert_recording_rules_last_evaluation_samples)
        by(group, recording) < 1
```

Monitoring is too complex…

1. **I noticed a pre-defined rule that matches no series**
2. I checked **scheduler_e2e_scheduling_duration_seconds_bucket** metric, but **it was missing** in TSDB

I google what is this metric and found the following:

Deprecated 2y ago…



Cleanup deprecated metrics #115209

Merged **k8s-ci-robot** merged 1 commit into `kubernetes:master` from `dgrisonne`

Conversation 16 · Commits 1 · Checks 0 · Files change

dgrisonnet commented on Jan 19, 2023 • edited

What type of PR is this?

/kind cleanup

What this PR does / why we need it:

Remove the following deprecated metrics:

- node_collector_evictions_number
- scheduler_e2e_scheduling_duration_seconds

Where it this rule comes from?
We import it from prometheus-operator…

```
{
    url:  "https://raw.githubusercontent.com/prometheus-operator/kube-prometheus/main/manifests/kubernetesControlPlane-prometheusRule.yaml",
    kind: "rules",
    charts: []string{
        "victoria-metrics-k8s-stack",
    },
```

Where they get it from?

From kuberentes-mixin … updated 6y ago

master kubernetes-mixin / rules / kube_scheduler.libsonnet

Go to file t

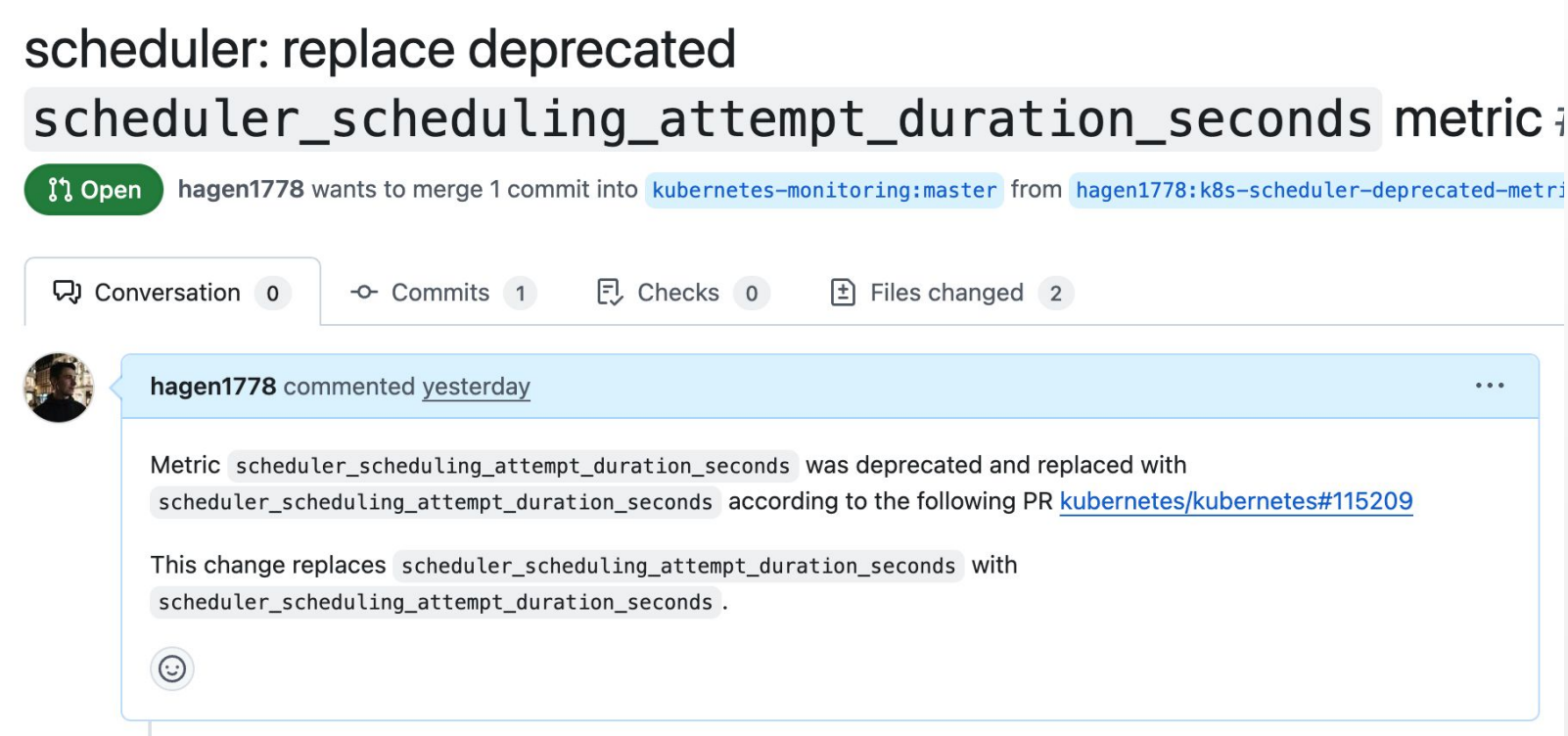
brancz Normalize file naming

303804a · 6 years ago History

Code Blame Raw

```
1   {
2     _config+:: {
3       kubeSchedulerSelector: 'job="kube-scheduler"',
4       podLabel: 'pod',
5     },
6
7     prometheusRules+:: {
8       groups+: [
9         {
10          name: 'kube-scheduler.rules',
11          rules: [
12            {
13              record: 'cluster_quantile:%s:histogram_quantile' % metric,
14              expr: |||
15                histogram_quantile(%(quantile)s, sum(rate(%(metric)s_bucket{%(kubeSchedulerSelector)s}[5m])) without(instance, %(podLabel)s))
16              ||| % ({ quantile: quantile, metric: metric } + $._config),
17              labels: {
18                quantile: quantile,
19              },
20            }
21            for quantile in ['0.99', '0.9', '0.5']
22            for metric in [
23              'scheduler_e2e_scheduling_duration_seconds',
24              'scheduler_scheduling_algorithm_duration_seconds',
25              'scheduler_binding_duration_seconds',
```

Do you still trust your monitoring?



VictoriaMetrics helps you understand your data

- Built-in view in VictoriaMetrics UI with a lot of useful features
- Shows amount of metrics and series
- Shows when and how many times metric was queried
- Allows finding metrics that are:
  - **never** queried
  - **occasionally** queried
  - **repeatedly** queried
- Allows finding alerting and recording rules that make no sense
- **Open Source**

The end