On the path to full Observability with OSS

David Kaltschmidt
@davkals

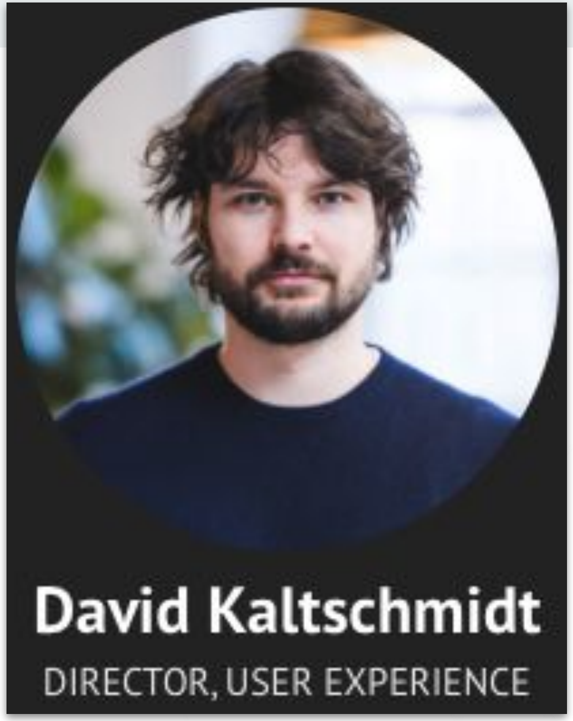
Kubecon 2018

I’m David

All things UX at Grafana Labs

If you click and are stuck, reach out to me.
david@grafana.com

Twitter: @davkals



Grafana Labs

Outline

- Quick Grafana intro
- Make an app observable
- Logging in detail

Grafana Labs

Grafana intro

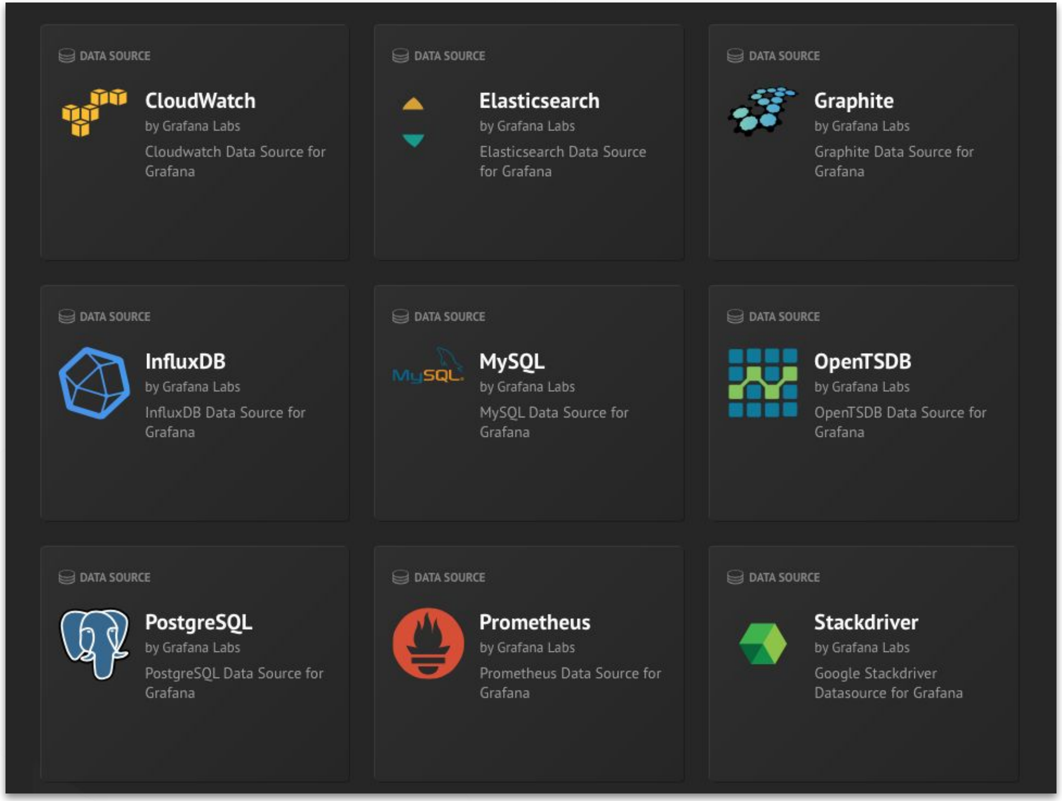
Grafana

Dashboarding solution

Observability platform

Grafana Labs

Unified way to look at data from different sources


DATA SOURCE
CloudWatch
by Grafana Labs
Cloudwatch Data Source for Grafana

DATA SOURCE
Elasticsearch
by Grafana Labs
Elasticsearch Data Source for Grafana

DATA SOURCE
Graphite
by Grafana Labs
Graphite Data Source for Grafana

DATA SOURCE
InfluxDB
by Grafana Labs
InfluxDB Data Source for Grafana

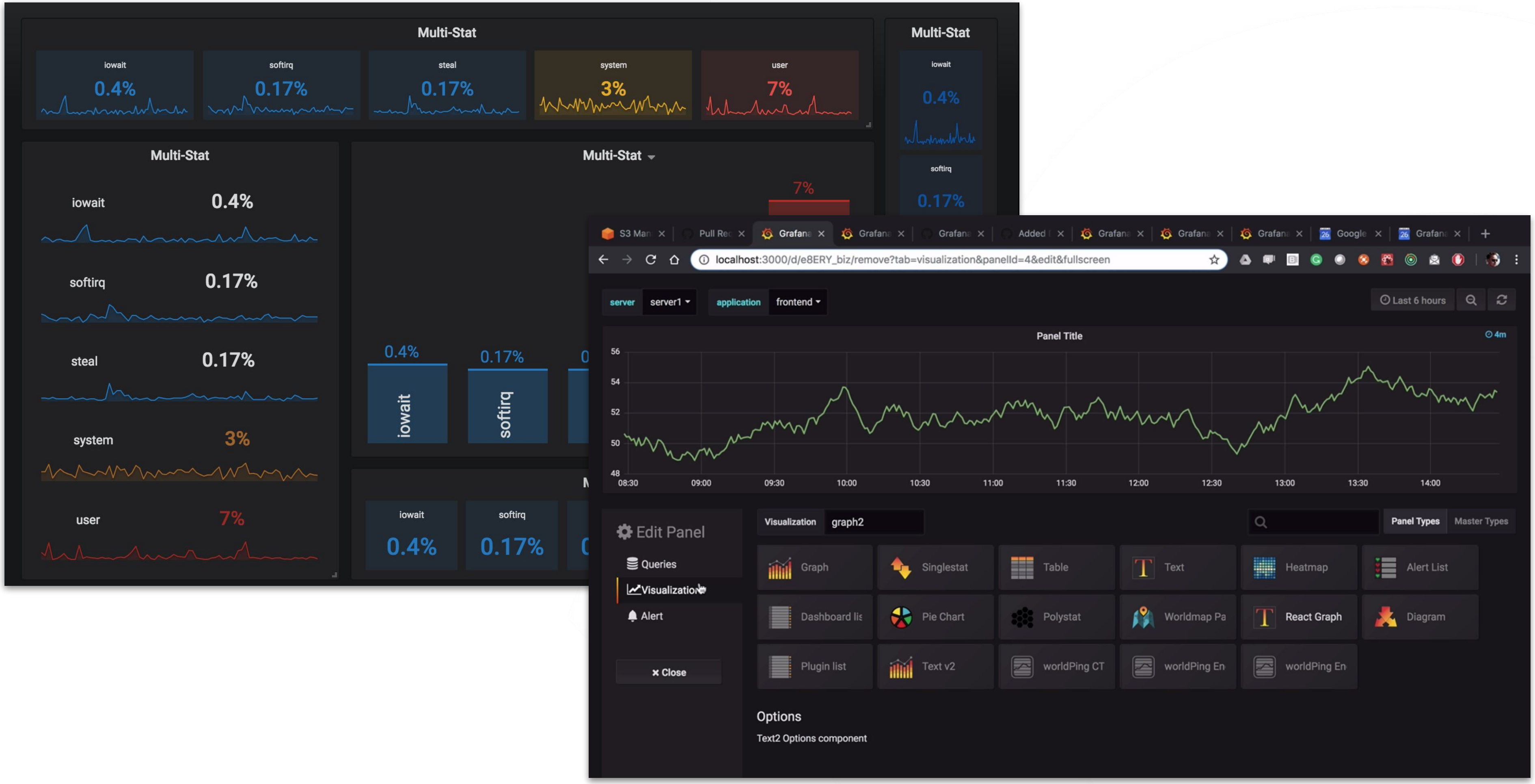
DATA SOURCE
MySQL
by Grafana Labs
MySQL Data Source for Grafana

DATA SOURCE
OpenTSDB
by Grafana Labs
OpenTSDB Data Source for Grafana

DATA SOURCE
PostgreSQL
by Grafana Labs
PostgreSQL Data Source for Grafana

DATA SOURCE
Prometheus
by Grafana Labs
Prometheus Data Source for Grafana

DATA SOURCE
Stackdriver
by Grafana Labs
Google Stackdriver Datasource for Grafana



localhost:3000/d/e8ERY_biz/remove?tab=visualization&panelId=4&edit&fullscreen

server: server1
application: frontend

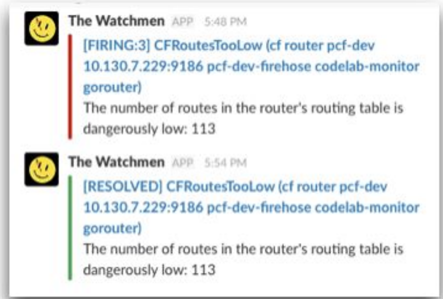
Visualization: graph2

Options

Text2 Options component

New graph panel controller to quickly iterate how to visualize

1. Alert

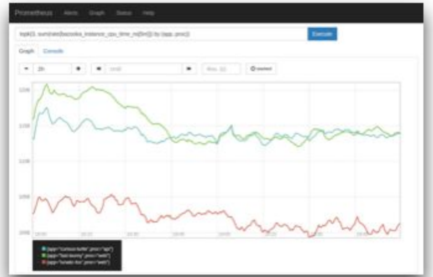


The Watchmen APP 5:48 PM
[FIRING:3] CFRoutesTooLow (cf router pcf-dev 10.130.7.229:9186 pcf-dev-firehose codelab-monitor gorouter)
The number of routes in the router's routing table is dangerously low: 113

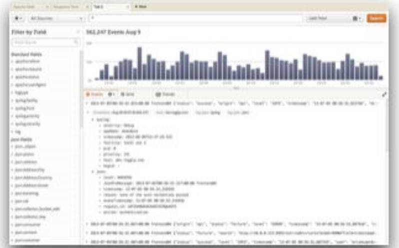
The Watchmen APP 5:54 PM
[RESOLVED] CFRoutesTooLow (cf router pcf-dev 10.130.7.229:9186 pcf-dev-firehose codelab-monitor gorouter)
The number of routes in the router's routing table is dangerously low: 113

2. Dashboard

3. Adhoc Query



4. Log Aggregation



5. Distributed Tracing

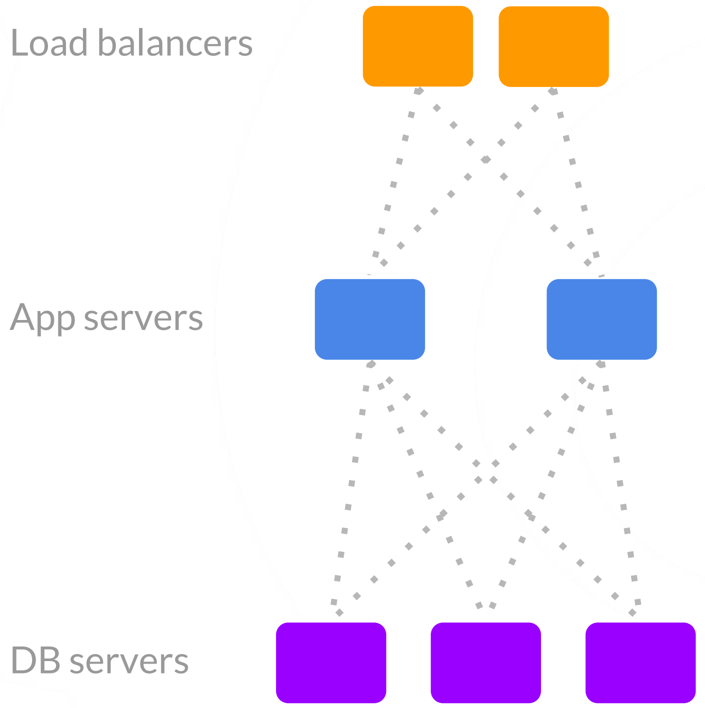
Fix!

Troubleshooting journey

Instrumenting an app

App

- Classic 3-tiered app
- Deployed in Kubernetes
- It’s running, but how is it doing?



Add instrumentation

- Make sure the app logs enough
- Add Prometheus client library for metrics
- Hook up Jaeger for distributed tracing

Structured Logging

```
logger = kitlog.NewLogfmtLogger(kitlog.NewSyncWriter(os.Stderr))
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
  since := time.Now()
  defer func() {
    logger.Log("level", "info", "msg", "query executed OK", "duration", time.Since(since))
  }()
  ...
  if fail {
    logger.Log("level", "error", "msg", "query lock timeout")
  }
  ...
})
```

Add instrumentation

- Make sure the app logs enough
- Add Prometheus client library for metrics
- Hook up Jaeger for distributed tracing

Metrics with Prometheus

```
requestDuration = promauto.NewHistogramVec(prometheus.HistogramOpts{
  Name:    "request_duration_seconds",
  Help:    "Time (in seconds) spent serving HTTP requests",
  Buckets: prometheus.DefBuckets,
}, []string{"method", "route", "status_code"})

func wrap(h http.HandlerFunc) http.HandlerFunc {
  return func(w http.ResponseWriter, r *http.Request) {
    m := httpsnoop.CaptureMetrics(h, w, r)
    requestDuration.WithLabelValues(r.Method, r.URL.Path,
strconv.Itoa(m.Code)).Observe(m.Duration.Seconds())
  }
}

http.HandleFunc("/", wrap(func(w http.ResponseWriter, r *http.Request) {}))
```

Add instrumentation

- Make sure the app logs enough
- Add Prometheus client library for metrics
- Hook up Jaeger for distributed tracing

Jaeger Tracing

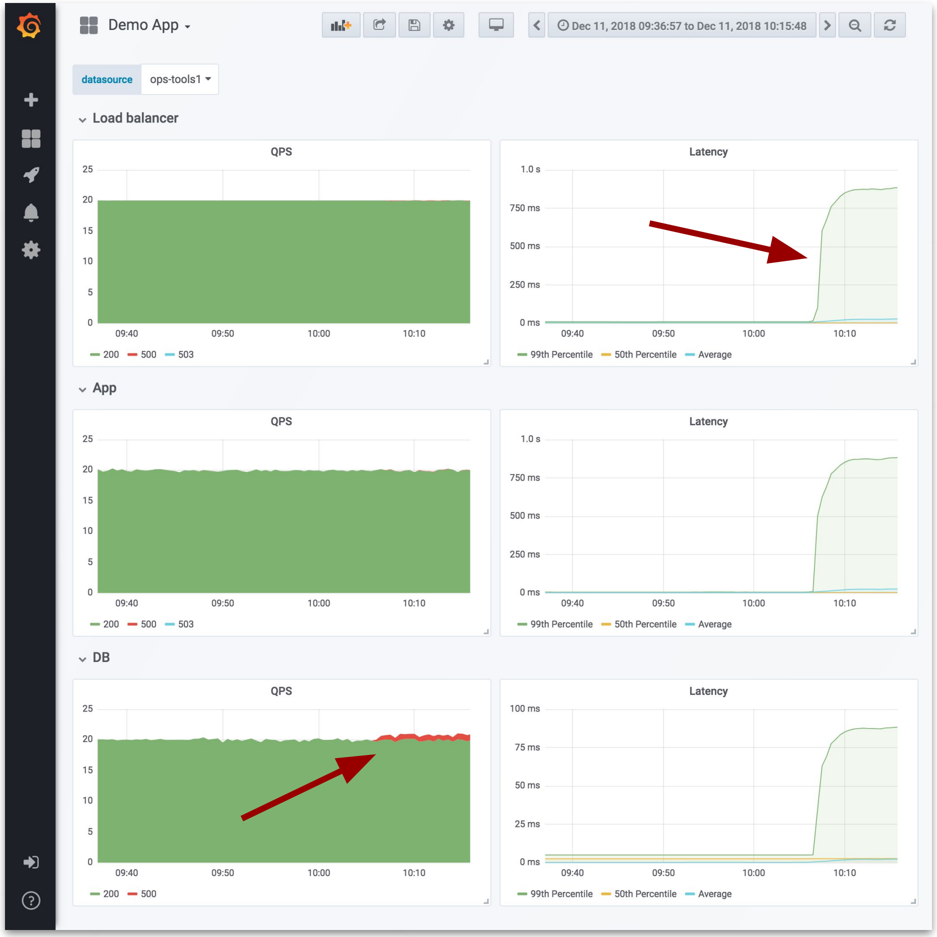
```
cfg, err := jaegercfg.FromEnv()
cfg.InitGlobalTracer("db")

http.HandleFunc("/", wrap(func(w http.ResponseWriter, r *http.Request) {}))

go func() {
  errc <- http.ListenAndServe(dbPort,
      nethttp.Middleware(opentracing.GlobalTracer(), http.DefaultServeMux))
}()
```

Bonus: Set up tools

- https://github.com/coreos/prometheus-operator Job to look after running Prometheus on Kubernetes and set of configs for all exporters you need to get Kubernetes metrics
- https://github.com/grafana/jsonnet-libs/tree/master/prometheus-ksonnet Our configs for running Prometheus, Alertmanager, Grafana together
- https://github.com/kubernetes-monitoring/kubernetes-mixin Joint project to unify and improve common alerts for Kubernetes

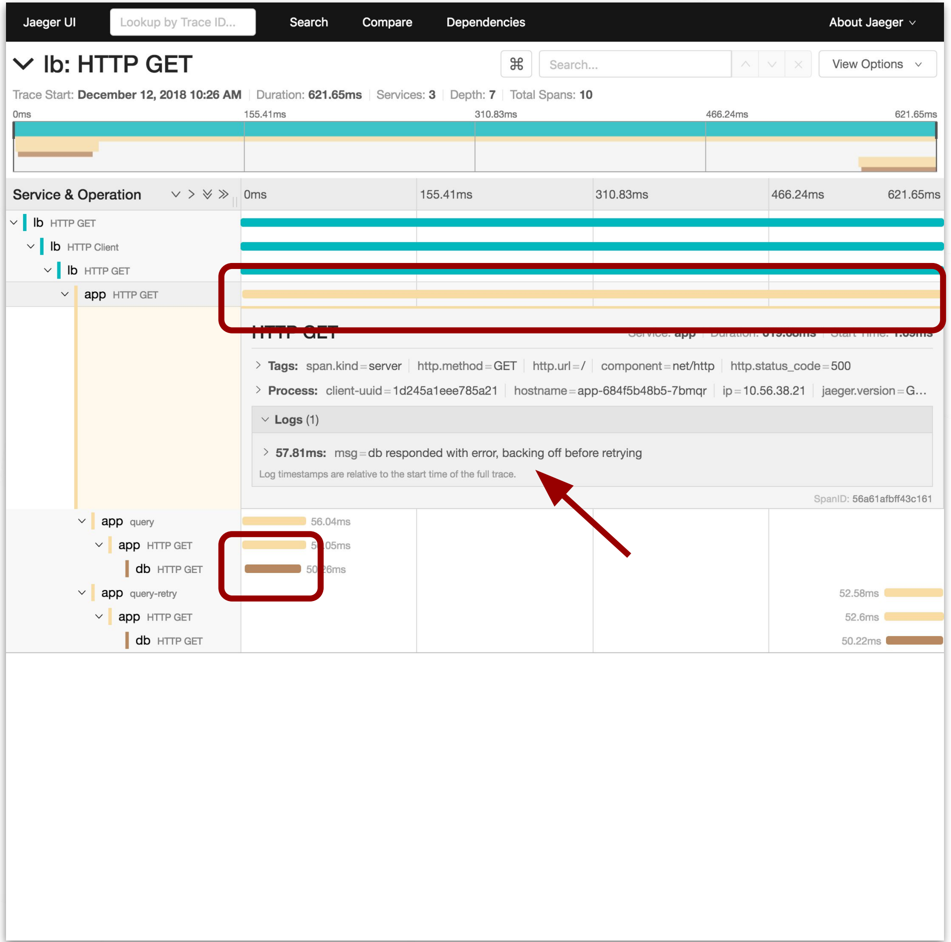
Live demo (screenshots follow)

- You've been paged because the p99 latency shot up from <10ms to >700ms
- RED method dashboard is ideal entrypoint to see health of the system
- Notice also DB error rates, luckily not bubbling up to user



RED method dashboard of the app

- Investigate latency issue first using Jaeger
- App is spending lots of time even though DB request returned quickly
- Root cause: backoff period was too high
- Idea for fix: lower backoff period



lb: HTTP GET

Trace Start: **December 12, 2018 10:26 AM** | Duration: **621.65ms** | Services: **3** | Depth: **7** | Total Spans: **10**

```
Service & Operation      0ms      155.41ms      310.83ms      466.24ms      621.65ms
lb HTTP GET
  lb HTTP Client
    lb HTTP GET
      app HTTP GET
```

HTTP GET — Service: **app**

Tags: span.kind = server | http.method = GET | http.url = / | component = net/http | http.status_code = 500

Process: client-uuid = 1d245a1eee785a21 | hostname = app-684f5b48b5-7bmqr | ip = 10.56.38.21 | jaeger.version = G...

Logs (1)

57.81ms: msg = db responded with error, backing off before retrying

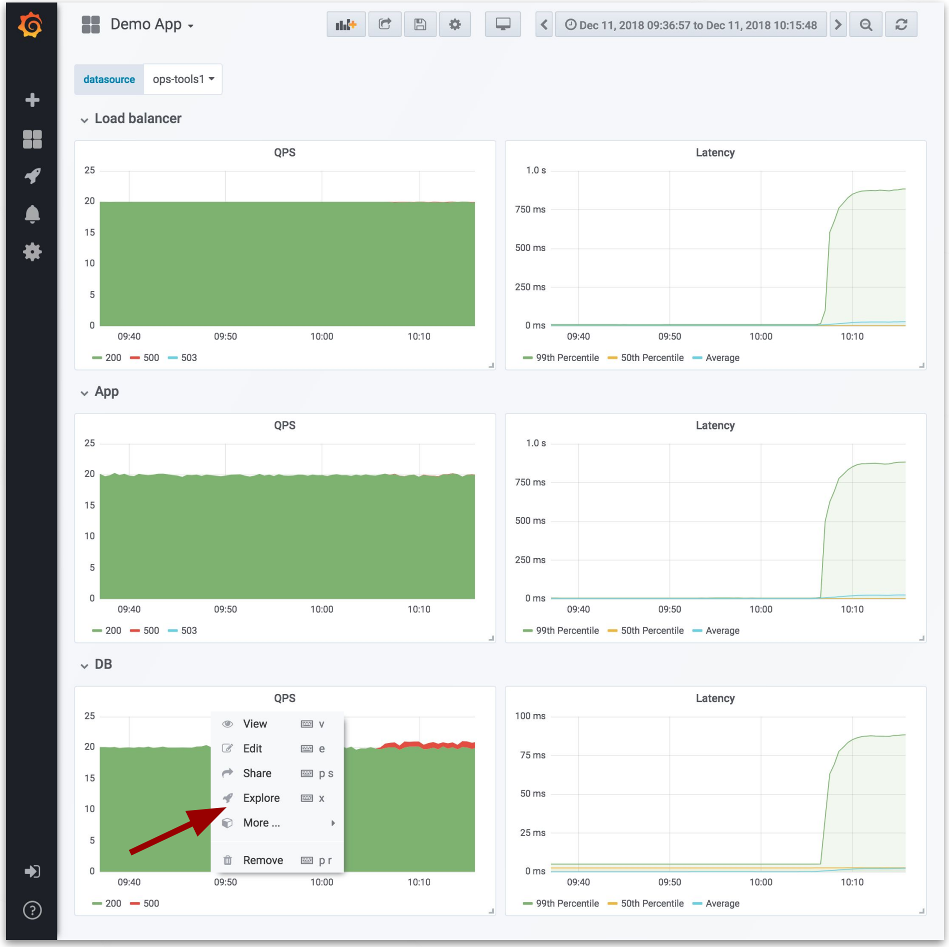
Log timestamps are relative to the start time of the full trace.

SpanID: 56a61afbff43c161

```
        app query            56.04ms
          app HTTP GET       5[illegible].05ms
            db HTTP GET      50[illegible]6ms
        app query-retry                                  52.58ms
          app HTTP GET                                   52.6ms
            db HTTP GET                                  50.22ms
```

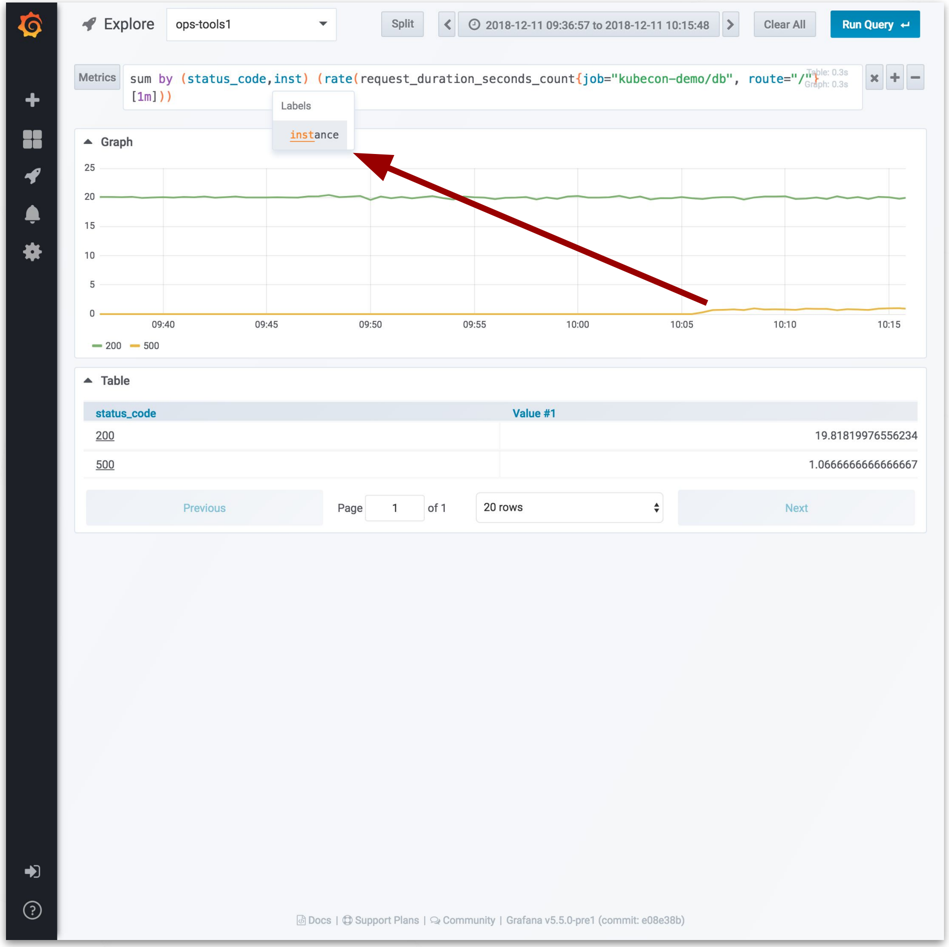
Debug latency issue with Jaeger

- Still need to investigate DB errors
- Jumping to Explore for query-driven troubleshooting



Jump to Explore from dashboard panel

- Explore pre-filled the query from the dashboard
- Interact with the query with smart tab completion
- Break down by “instance” to check which DB instance is producing errors

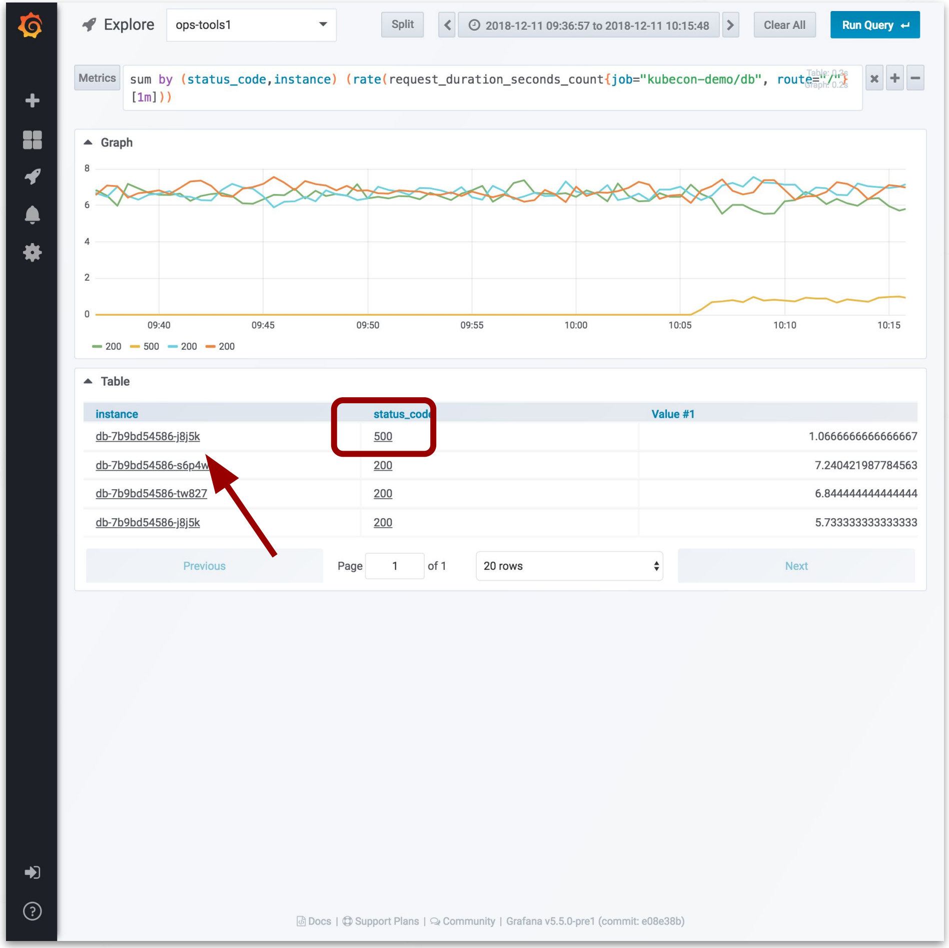


```
sum by (status_code,inst) (rate(request_duration_seconds_count{job="kubecon-demo/db", route="/"}
[1m]))
```

status_code	Value #1
200	19.81819976556234
500	1.0666666666666667

Explore for query interaction

- Breakdown by instance shows single instance producing 500s (error status code)
- Click on instance label to narrow down further

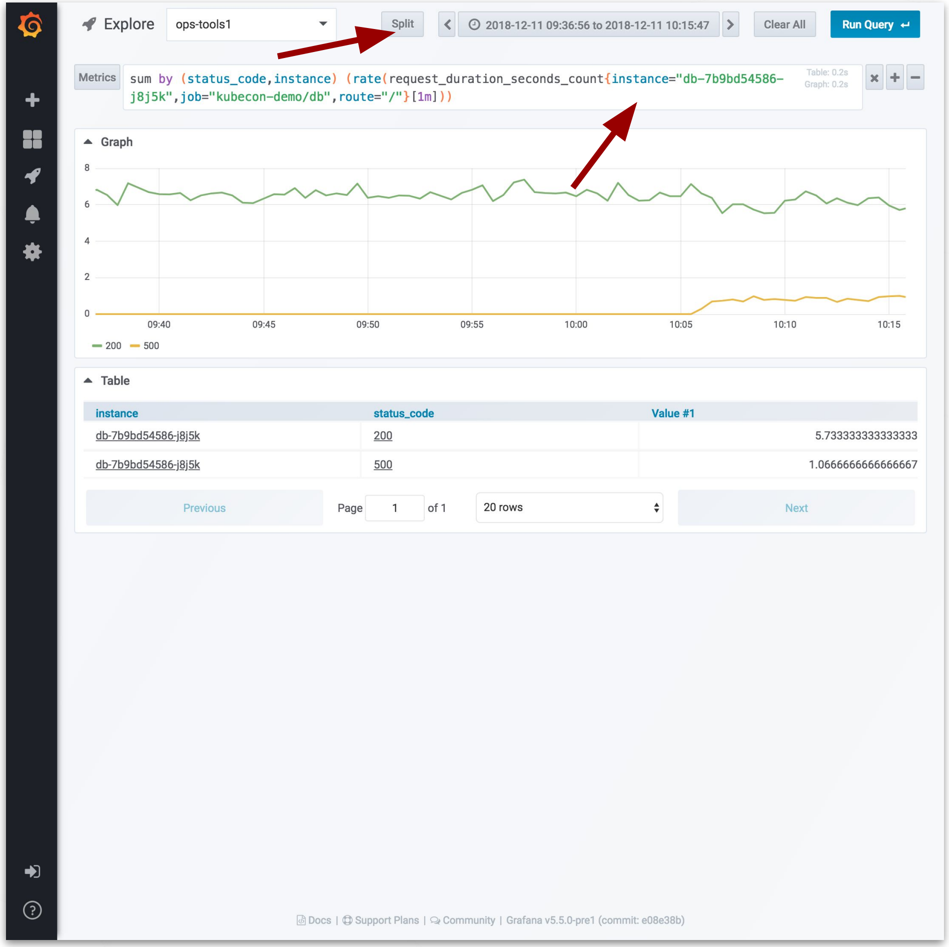

Explore
ops-tools1
Split
2018-12-11 09:36:57 to 2018-12-11 10:15:48
Clear All
Run Query
Metrics
Graph
Table
Previous
Page 1 of 1
20 rows
Next
Docs | Support Plans | Community | Grafana v5.5.0-pre1 (commit: e08e38b)

```
sum by (status_code,instance) (rate(request_duration_seconds_count{job="kubecon-demo/db", route="/"}
[1m]))
```

instance	status_code	Value #1
db-7b9bd54586-j8j5k	500	1.0666666666666667
db-7b9bd54586-s6p4w	200	7.240421987784563
db-7b9bd54586-tw827	200	6.844444444444444
db-7b9bd54586-j8j5k	200	5.733333333333333

Explore for query interaction

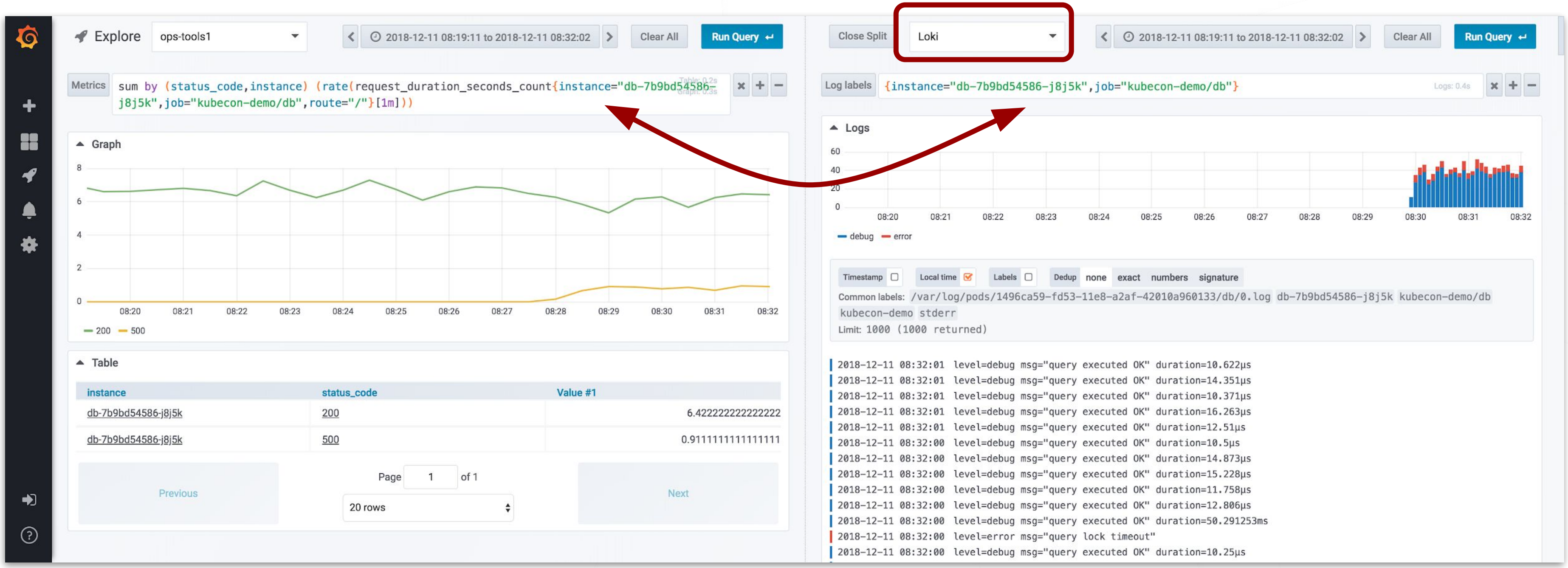
- Instance label is now part of the query selector
- We've isolated the DB instance and see only its metrics
- Now we can split the view and select the logging datasource



```
sum by (status_code,instance) (rate(request_duration_seconds_count{instance="db-7b9bd54586-j8j5k",job="kubecon-demo/db",route="/"}[1m]))
```

instance	status_code	Value #1
db-7b9bd54586-j8j5k	200	5.733333333333333
db-7b9bd54586-j8j5k	500	1.0666666666666667

Explore for query interaction



```
sum by (status_code,instance) (rate(request_duration_seconds_count{instance="db-7b9bd54586-j8j5k",job="kubecon-demo/db",route="/"}[1m]))
```

```
{instance="db-7b9bd54586-j8j5k",job="kubecon-demo/db"}
```

instance	status_code	Value #1
db-7b9bd54586-j8j5k	200	6.4222222222222222
db-7b9bd54586-j8j5k	500	0.91111111111111111

Common labels: /var/log/pods/1496ca59-fd53-11e8-a2af-42010a960133/db/0.log db-7b9bd54586-j8j5k kubecon-demo/db kubecon-demo stderr

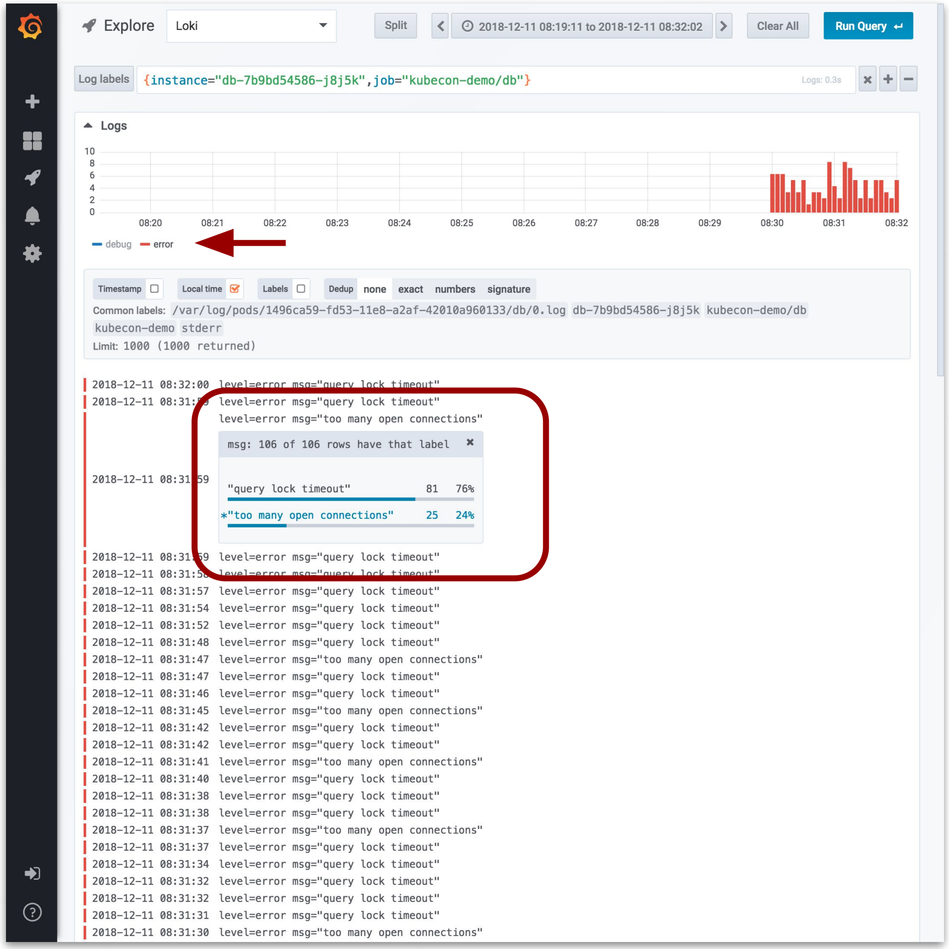
Limit: 1000 (1000 returned)

```
2018-12-11 08:32:01  level=debug msg="query executed OK" duration=10.622µs
2018-12-11 08:32:01  level=debug msg="query executed OK" duration=14.351µs
2018-12-11 08:32:01  level=debug msg="query executed OK" duration=10.371µs
2018-12-11 08:32:01  level=debug msg="query executed OK" duration=16.263µs
2018-12-11 08:32:01  level=debug msg="query executed OK" duration=12.51µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=10.5µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=14.873µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=15.228µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=11.758µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=12.806µs
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=50.291253ms
2018-12-11 08:32:00  level=error msg="query lock timeout"
2018-12-11 08:32:00  level=debug msg="query executed OK" duration=10.25µs
```

- Right side switch over a logging datasource
- Logging query retains the Prometheus query labels to select the log stream

Metrics and logs side-by-side

- Filter for log level error using the graph legend
- Ad-hoc stats on structured log fields
- Root cause found: “Too many open connections”
- Idea for fix: more DB replicas, or connection pooling



```
{instance="db-7b9bd54586-j8j5k",job="kubecon-demo/db"}
```

Common labels: /var/log/pods/1496ca59-fd53-11e8-a2af-42010a960133/db/0.log db-7b9bd54586-j8j5k kubecon-demo/db kubecon-demo stderr

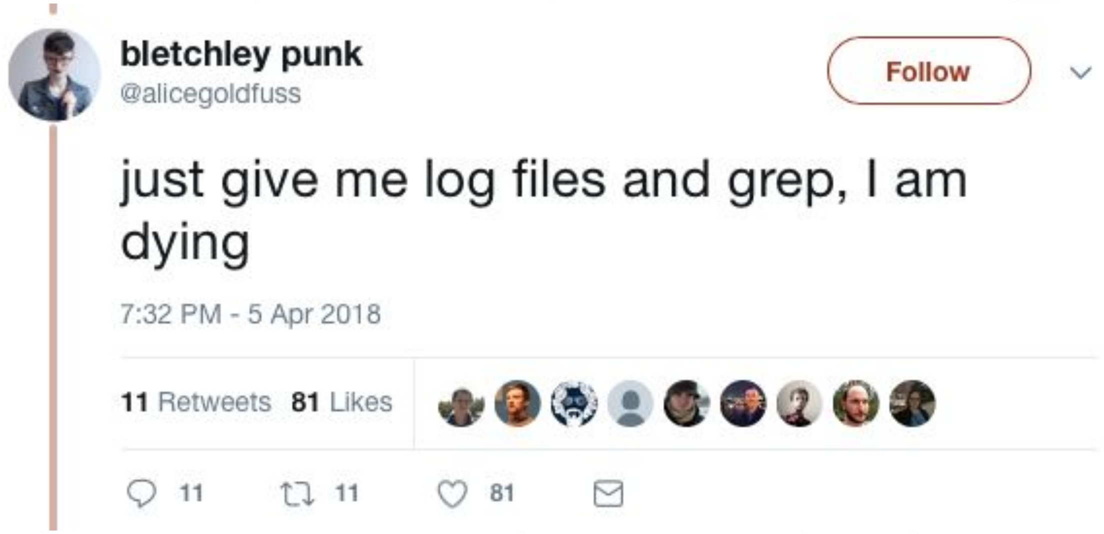
Limit: 1000 (1000 returned)

```
2018-12-11 08:32:00  level=error msg="query lock timeout"
2018-12-11 08:31:59  level=error msg="query lock timeout"
                     level=error msg="too many open connections"
                     msg: 106 of 106 rows have that label
2018-12-11 08:31:59
                     "query lock timeout"              81   76%
                    *"too many open connections"       25   24%
2018-12-11 08:31:59  level=error msg="query lock timeout"
2018-12-11 08:31:58  level=error msg="query lock timeout"
2018-12-11 08:31:57  level=error msg="query lock timeout"
2018-12-11 08:31:54  level=error msg="query lock timeout"
2018-12-11 08:31:52  level=error msg="query lock timeout"
2018-12-11 08:31:48  level=error msg="query lock timeout"
2018-12-11 08:31:47  level=error msg="too many open connections"
2018-12-11 08:31:47  level=error msg="query lock timeout"
2018-12-11 08:31:46  level=error msg="query lock timeout"
2018-12-11 08:31:45  level=error msg="too many open connections"
2018-12-11 08:31:42  level=error msg="query lock timeout"
2018-12-11 08:31:42  level=error msg="query lock timeout"
2018-12-11 08:31:41  level=error msg="too many open connections"
2018-12-11 08:31:40  level=error msg="query lock timeout"
2018-12-11 08:31:38  level=error msg="query lock timeout"
2018-12-11 08:31:38  level=error msg="query lock timeout"
2018-12-11 08:31:37  level=error msg="too many open connections"
2018-12-11 08:31:37  level=error msg="query lock timeout"
2018-12-11 08:31:34  level=error msg="query lock timeout"
2018-12-11 08:31:32  level=error msg="query lock timeout"
2018-12-11 08:31:32  level=error msg="query lock timeout"
2018-12-11 08:31:31  level=error msg="query lock timeout"
2018-12-11 08:31:30  level=error msg="too many open connections"
```

Explore for query interaction

Grafana logging in detail

Goal: Keeping it simple


bletchley punk
@alicegoldfuss

just give me log files and grep, I am dying

7:32 PM - 5 Apr 2018

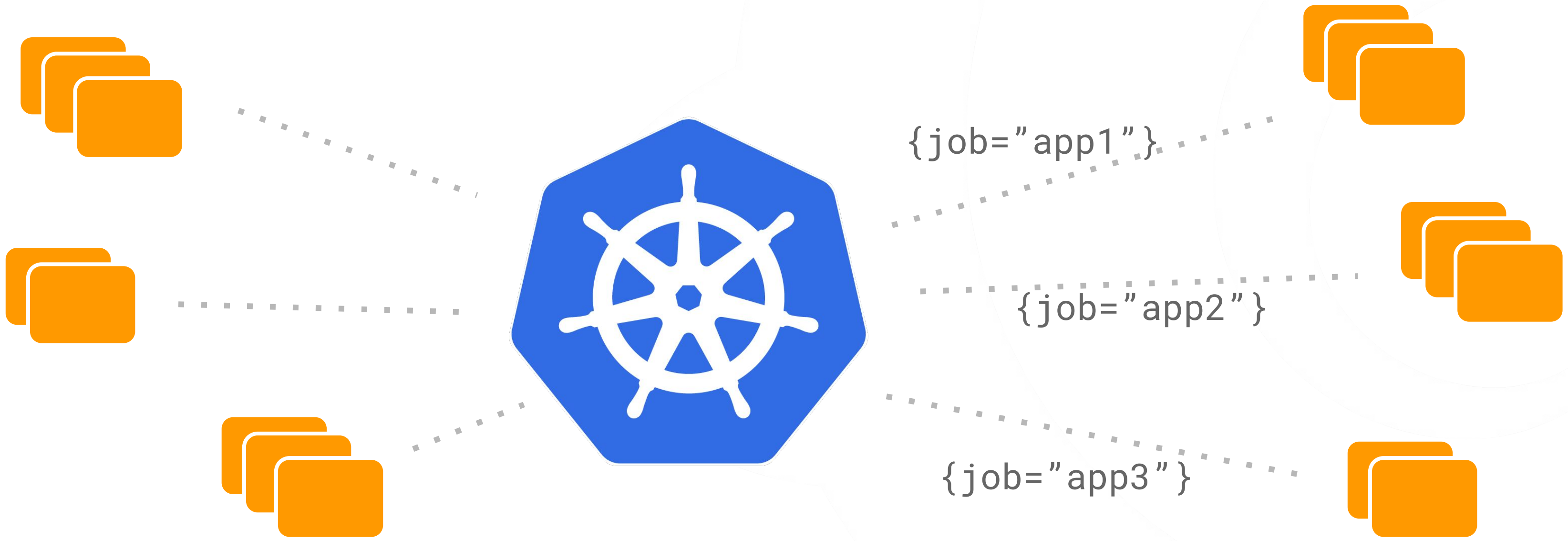
11 Retweets 81 Likes

https://twitter.com/alicegoldfuss/status/981947777256079360

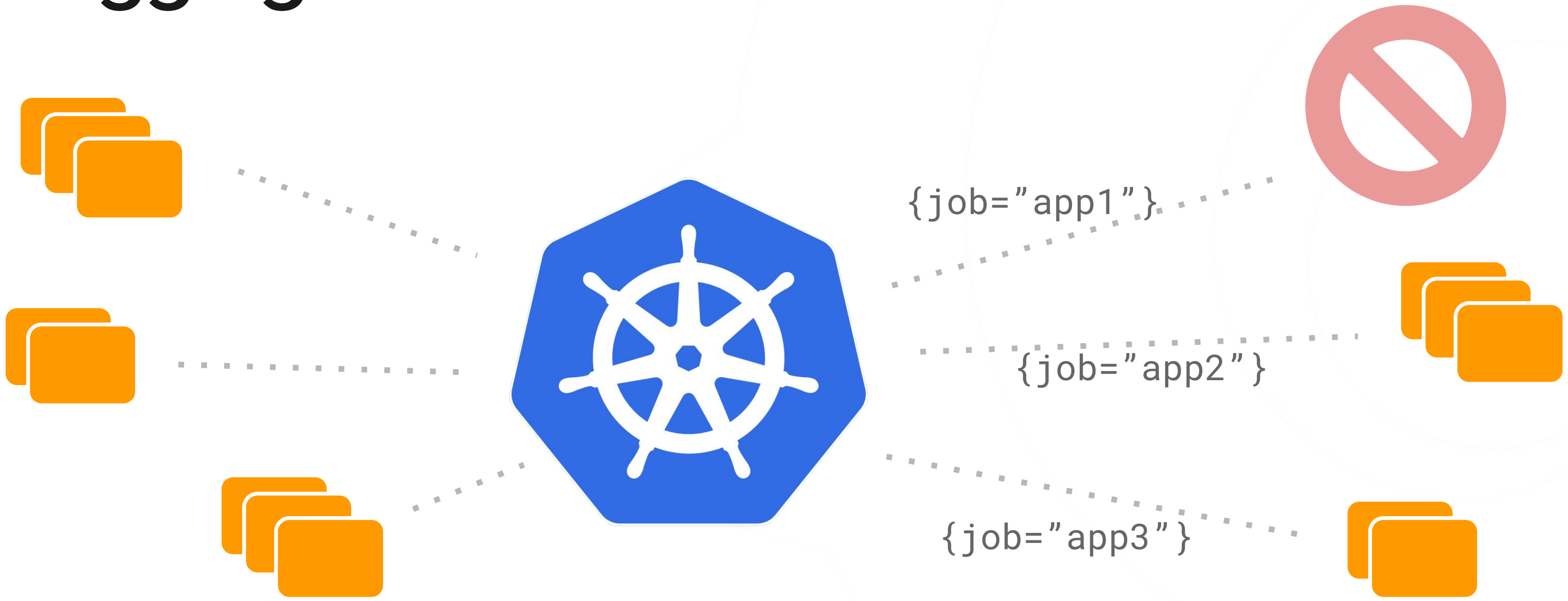
More goals

- Logs should be **cheap**!
- We found existing solutions are **hard to scale**
- We didn’t need **full text indexing**
- **Do ad-hoc analysis in the browser**

Logging for Kubernetes

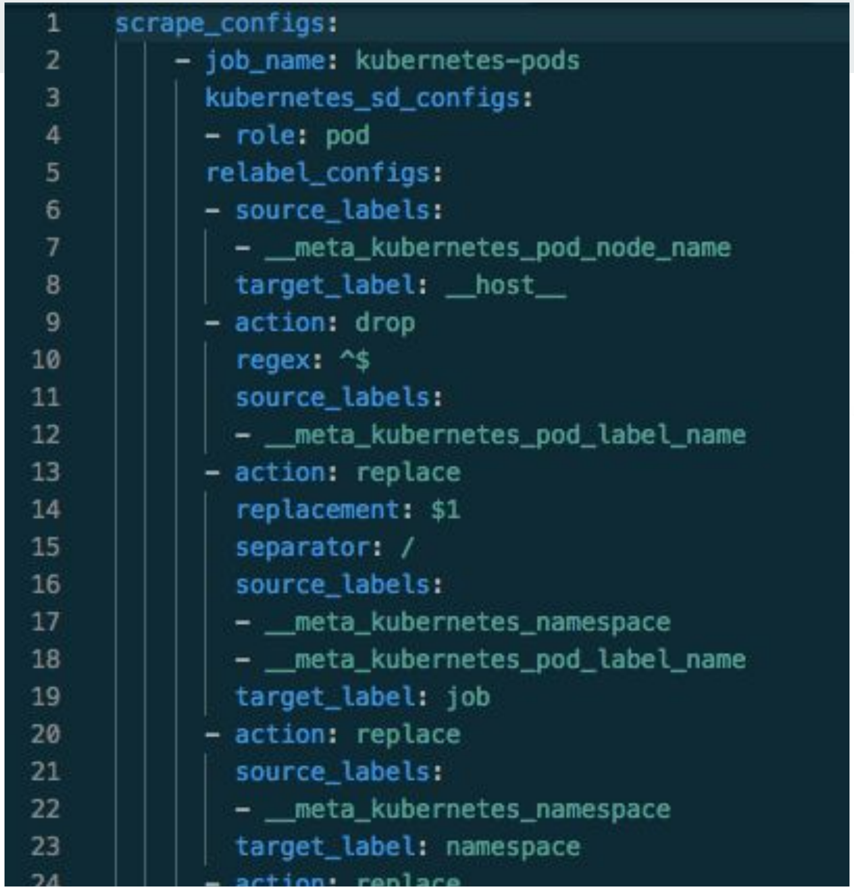


Logging for Kubernetes (2)



Like Prometheus, but for logs

- Prometheus-style service discovery of logging targets
- Labels are indexed as metadata, e.g.: {job=”app1”}



```
scrape_configs:
    - job_name: kubernetes-pods
      kubernetes_sd_configs:
      - role: pod
      relabel_configs:
      - source_labels:
        - __meta_kubernetes_pod_node_name
        target_label: __host__
      - action: drop
        regex: ^$
        source_labels:
        - __meta_kubernetes_pod_label_name
      - action: replace
        replacement: $1
        separator: /
        source_labels:
        - __meta_kubernetes_namespace
        - __meta_kubernetes_pod_label_name
        target_label: job
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: namespace
      - action: replace
```

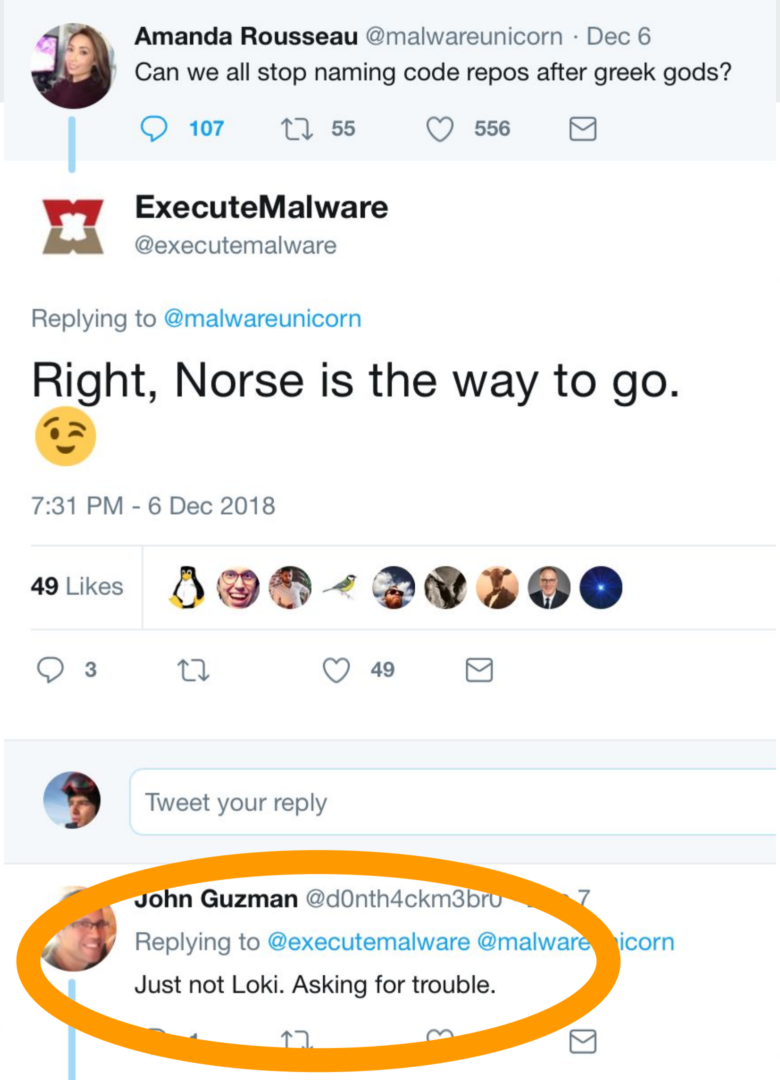
Introducing Loki

- Grafana's log aggregation service
- OSS and hosted

Grafana Labs

Introducing Loki

https://twitter.com/executemalware/status/1070747577811906560



Amanda Rousseau @malwareunicorn · Dec 6
Can we all stop naming code repos after greek gods?

107 55 556

ExecuteMalware
@executemalware

Replying to @malwareunicorn

Right, Norse is the way to go. 😉

7:31 PM - 6 Dec 2018

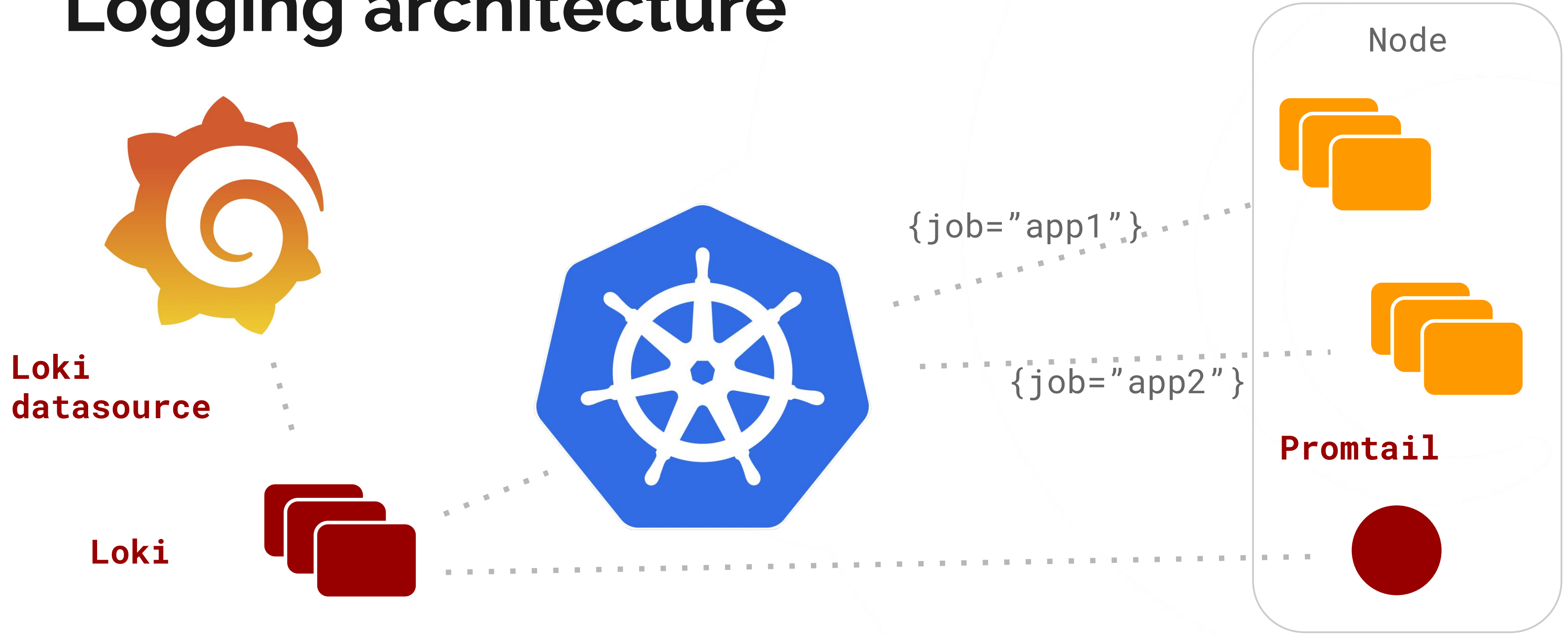
49 Likes

3 49

Tweet your reply

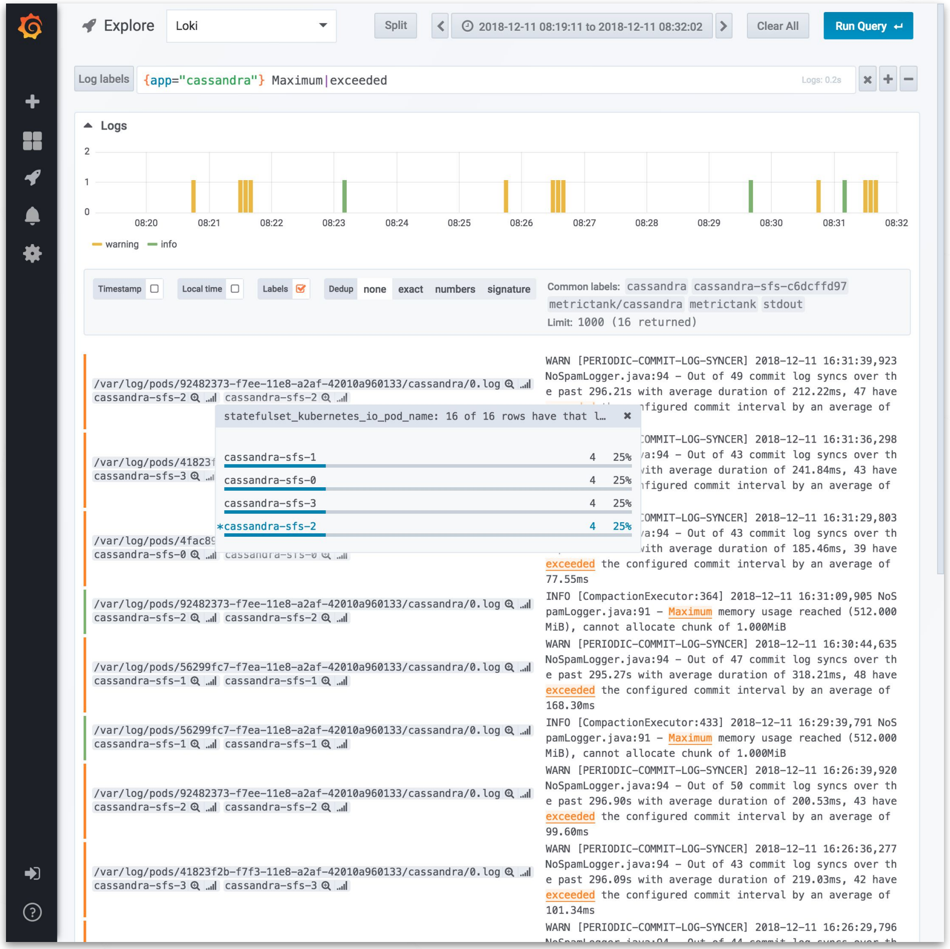
John Guzman @d0nth4ckm3br0 · Dec 7
Replying to @executemalware @malwareunicorn
Just not Loki. Asking for trouble.

Logging architecture



See Loki logs inside Grafana

- New builtin Loki datasource
- Prometheus-style stream selector
- Regexp filtering by the backend
- Simple UI:
    - no paging
    - return and render 1000 rows by default
    - Use the power of Cmd+F



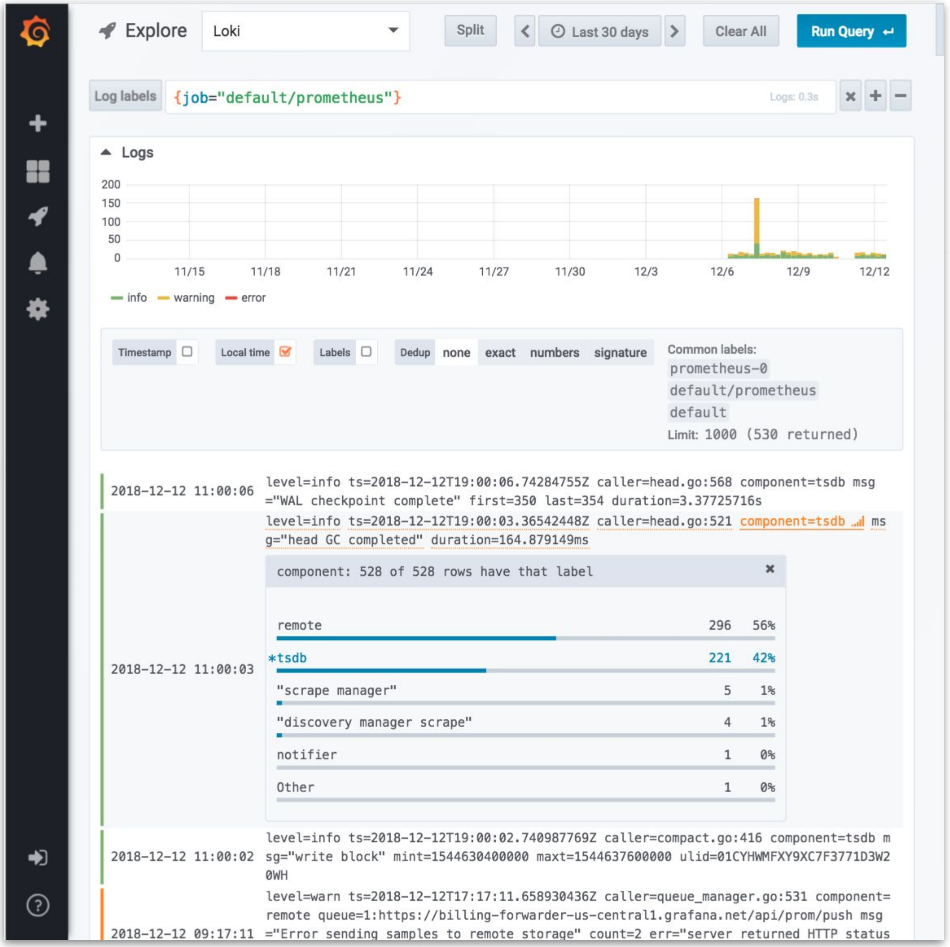
```
{app="cassandra"} Maximum|exceeded
```

Common labels: cassandra cassandra-sfs-c6dcffd97 metrictank/cassandra metrictank stdout
Limit: 1000 (16 returned)

```
statefulset_kubernetes_io_pod_name: 16 of 16 rows have that l…
cassandra-sfs-1     4  25%
cassandra-sfs-0     4  25%
cassandra-sfs-3     4  25%
*cassandra-sfs-2    4  25%
```

```
/var/log/pods/92482373-f7ee-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-2  cassandra-sfs-2
WARN [PERIODIC-COMMIT-LOG-SYNCER] 2018-12-11 16:31:39,923
NoSpamLogger.java:94 - Out of 49 commit log syncs over th
e past 296.21s with average duration of 212.22ms, 47 have
nfigured commit interval by an average of

/var/log/pods/41823f
cassandra-sfs-3
COMMIT-LOG-SYNCER] 2018-12-11 16:31:36,298
a:94 - Out of 43 commit log syncs over th
with average duration of 241.84ms, 43 have
nfigured commit interval by an average of

/var/log/pods/4fac8
cassandra-sfs-0  cassandra-sfs-0
COMMIT-LOG-SYNCER] 2018-12-11 16:31:29,803
a:94 - Out of 43 commit log syncs over th
with average duration of 185.46ms, 39 have
exceeded the configured commit interval by an average of
77.55ms

/var/log/pods/92482373-f7ee-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-2  cassandra-sfs-2
INFO [CompactionExecutor:364] 2018-12-11 16:31:09,905 NoS
pamLogger.java:91 - Maximum memory usage reached (512.000
MiB), cannot allocate chunk of 1.000MiB

/var/log/pods/56299fc7-f7ea-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-1  cassandra-sfs-1
WARN [PERIODIC-COMMIT-LOG-SYNCER] 2018-12-11 16:30:44,635
NoSpamLogger.java:94 - Out of 47 commit log syncs over th
e past 295.27s with average duration of 318.21ms, 48 have
exceeded the configured commit interval by an average of
168.30ms

/var/log/pods/56299fc7-f7ea-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-1  cassandra-sfs-1
INFO [CompactionExecutor:433] 2018-12-11 16:29:39,791 NoS
pamLogger.java:91 - Maximum memory usage reached (512.000
MiB), cannot allocate chunk of 1.000MiB

/var/log/pods/92482373-f7ee-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-2  cassandra-sfs-2
WARN [PERIODIC-COMMIT-LOG-SYNCER] 2018-12-11 16:26:39,920
NoSpamLogger.java:94 - Out of 50 commit log syncs over th
e past 296.90s with average duration of 200.53ms, 43 have
exceeded the configured commit interval by an average of
99.60ms

/var/log/pods/41823f2b-f7f3-11e8-a2af-42010a960133/cassandra/0.log
cassandra-sfs-3  cassandra-sfs-3
WARN [PERIODIC-COMMIT-LOG-SYNCER] 2018-12-11 16:26:36,277
NoSpamLogger.java:94 - Out of 43 commit log syncs over th
e past 296.09s with average duration of 219.03ms, 42 have
exceeded the configured commit interval by an average of
101.34ms

WARN [PERIODIC-COMMIT-LOG-SYNCER] 2018-12-11 16:26:29,796
```

- Various dedup options
- In-browser line parsing support for JSON and logfmt
- Ad-hoc stats across returned results (up to 1000 rows by default)
- Coming soon: ad-hoc graphs based on parsed numbers



Log labels `{job="default/prometheus"}` Logs: 0.3s

Common labels: prometheus-0 default/prometheus default
Limit: 1000 (530 returned)

```
2018-12-12 11:00:06  level=info ts=2018-12-12T19:00:06.74284755Z caller=head.go:568 component=tsdb msg
                     ="WAL checkpoint complete" first=350 last=354 duration=3.37725716s
2018-12-12 11:00:03  level=info ts=2018-12-12T19:00:03.36542448Z caller=head.go:521 component=tsdb ms
                     g="head GC completed" duration=164.879149ms
```

component: 528 of 528 rows have that label

remote	296	56%
*tsdb	221	42%
"scrape manager"	5	1%
"discovery manager scrape"	4	1%
notifier	1	0%
Other	1	0%

```
2018-12-12 11:00:02  level=info ts=2018-12-12T19:00:02.740987769Z caller=compact.go:416 component=tsdb m
                     sg="write block" mint=1544630400000 maxt=1544637600000 ulid=01CYHWMFXY9XC7F3771D3W2
                     0WH
2018-12-12 09:17:11  level=warn ts=2018-12-12T17:17:11.658930436Z caller=queue_manager.go:531 component=
                     remote queue=1:https://billing-forwarder-us-central1.grafana.net/api/prom/push msg
                     ="Error sending samples to remote storage" count=2 err="server returned HTTP status
```

See Loki logs inside Grafana

Release Loki

Loki OSS:
https://github.com/grafana/loki

Hosted Loki:
https://grafana.com/loki

All You Can Log trial
free until Q2, 2019

Grafana Labs

Enable Explore UI (BETA)

Logging UI is behind feature flag. To enable, edit Grafana config.ini file

```
[explore]
enabled = true
```

Explore will be released in Grafana v6.0 (Feb 2019)

Loki can be used today

Feedback welcome: @davkals or david@grafana.com

Integrate Tracing

- Associate traces with logs and metrics
- Labels and Exemplars FTW
- Aiming for Q2 2019

One last thing...

LOS ANGELES

GrafanaCon

Feb 25-26 2019

Expires Dec 19

https://www.grafanacon.org/2019/

Discount $100 off: KUBECON-LOKI-GRAF

Tack for listening

UX feedback to
david@grafana.com
@davkals

Tack for listening

UX feedback to
david@grafana.com
@davkals