Active Benchmarking RGW

Marcel Lauhoff | Staff Software Engineer
October, 2025

Active Benchmarking

With *active benchmarking,* you analyze performance while the benchmark is still running (...), using other tools. You can confirm that the benchmark tests what you intend it to, and that you understand what this is. Data becomes Information.

(https://www.brendangregg.com/activebenchmarking.html)

CL:SO

Stack

1. Workload Generator
2. Observability Stack
3. System Under Test

k6

Prometheus + Grafana

mtail, ceph-exporter

RGW

k6

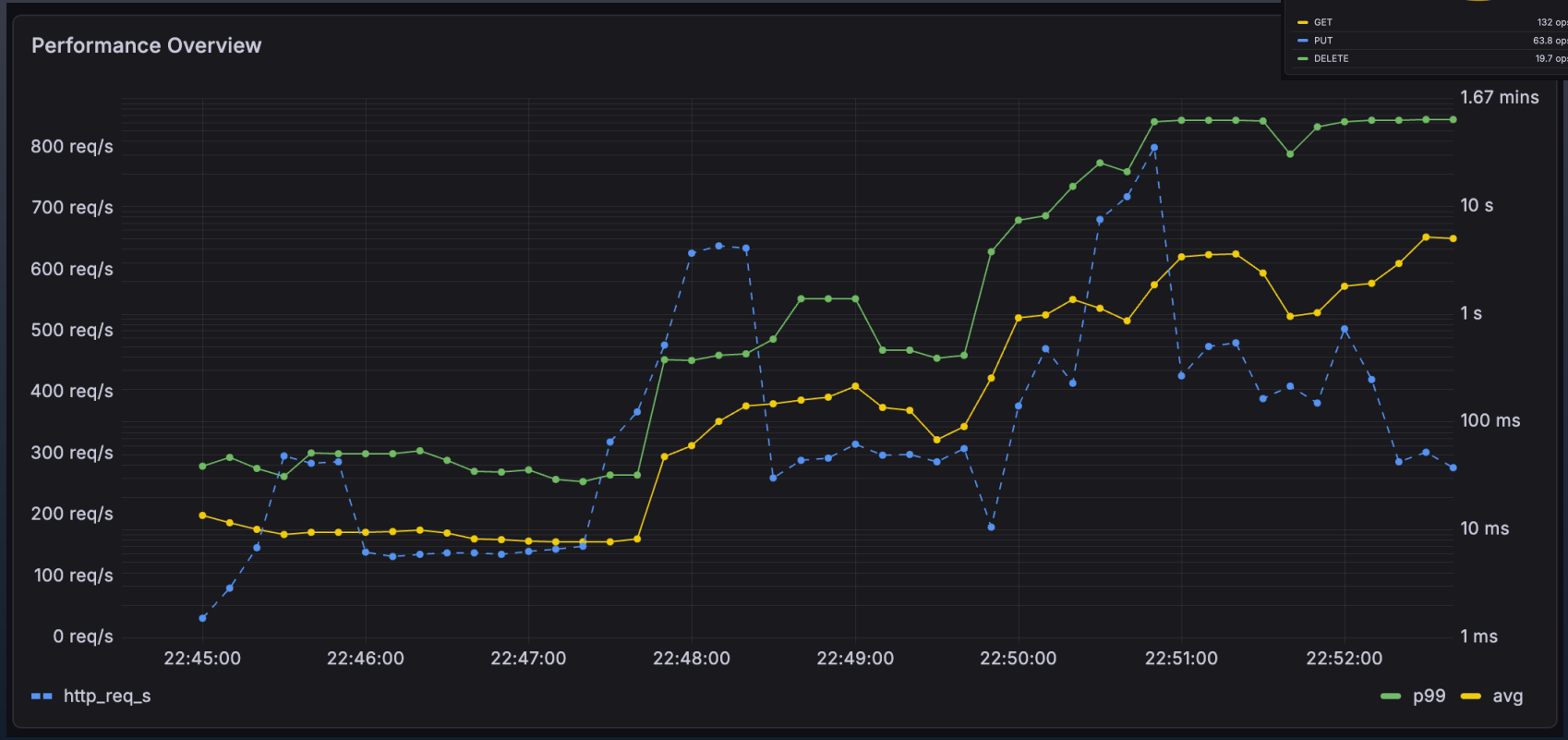
Run JavaScript
in a loop
with instrumentation
at scale

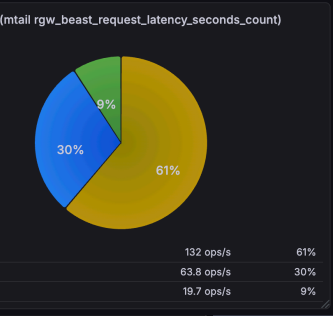
CL'SO

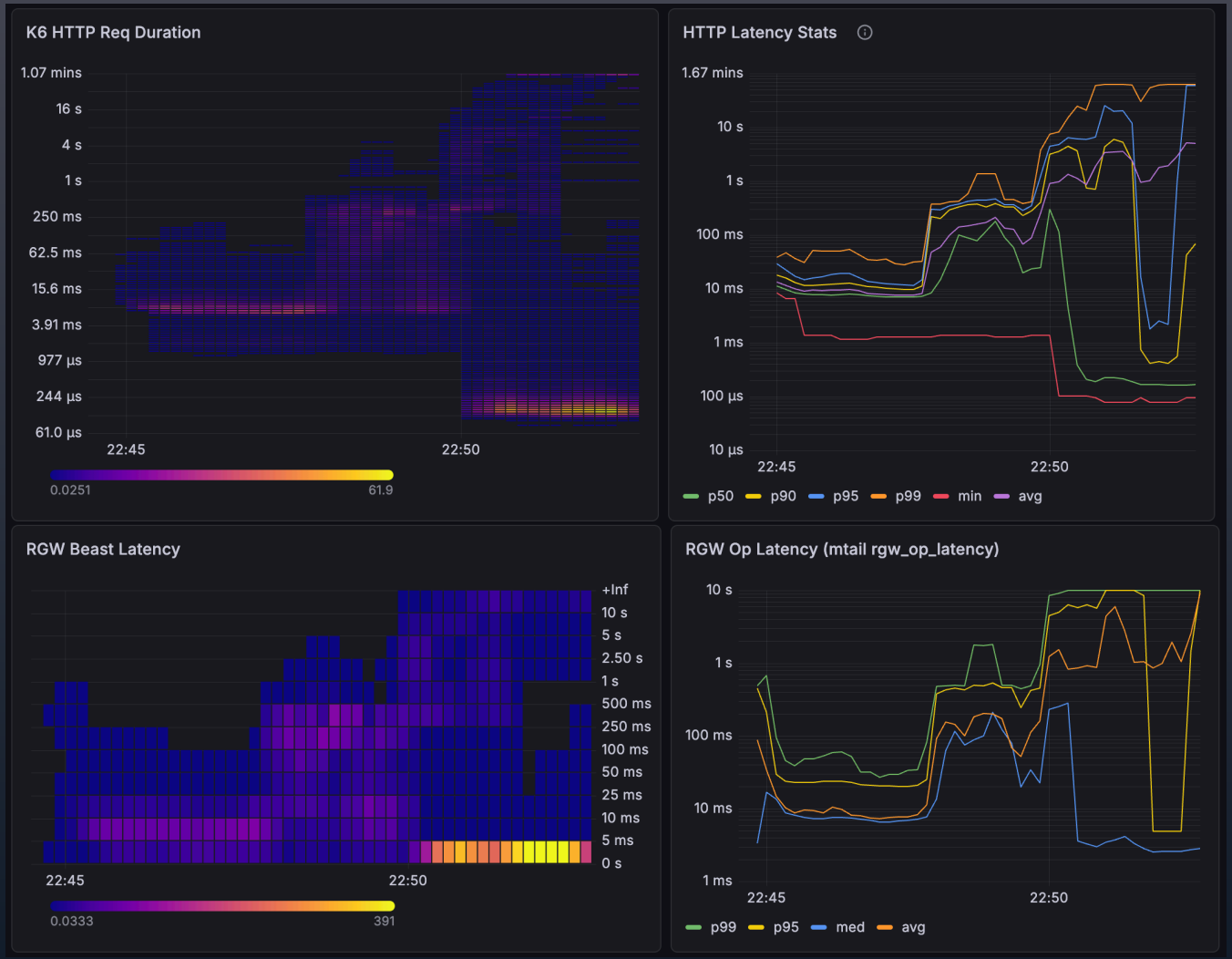
Let’s Look at Dashboards

CL'SO

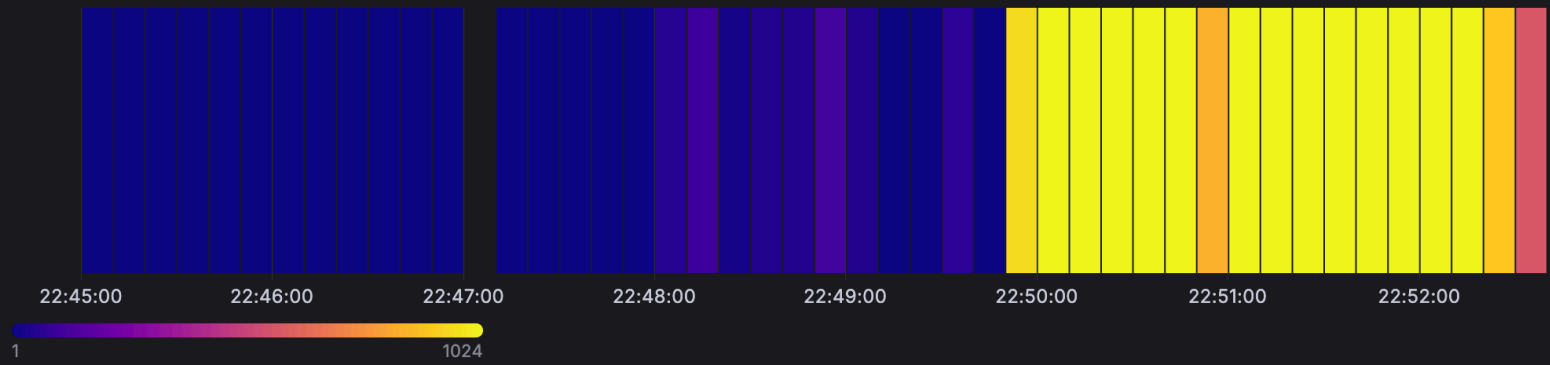
Load: Light -> Marginal -> Saturated



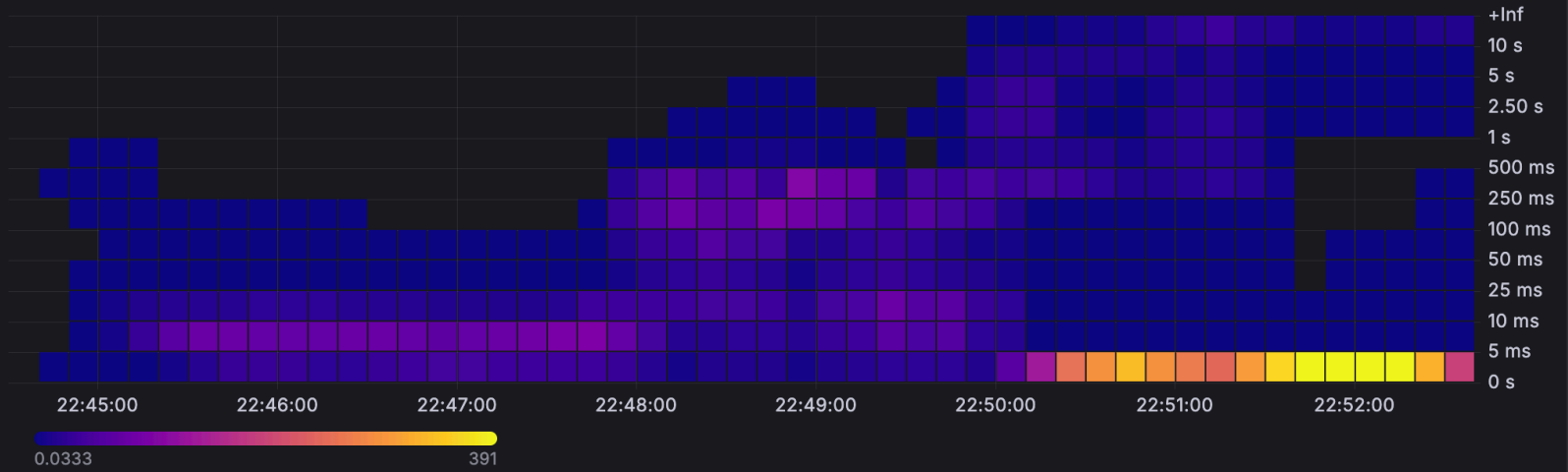


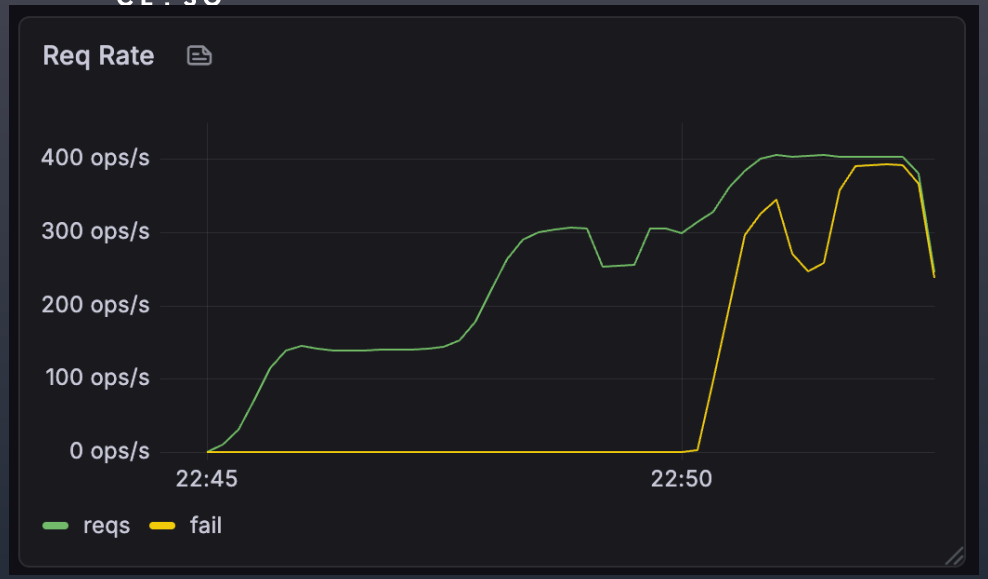

K6 HTTP Req Duration
1.07 mins
16 s
4 s
1 s
250 ms
62.5 ms
15.6 ms
3.91 ms
977 µs
244 µs
61.0 µs
22:45
22:50
0.0251
61.9
HTTP Latency Stats
1.67 mins
10 s
1 s
100 ms
10 ms
1 ms
100 µs
10 µs
22:45
22:50
p50
p90
p95
p99
min
avg
RGW Beast Latency
+Inf
10 s
5 s
2.50 s
1 s
500 ms
250 ms
100 ms
50 ms
25 ms
10 ms
5 ms
0 s
22:45
22:50
0.0333
391
RGW Op Latency (mtail rgw_op_latency)
10 s
1 s
100 ms
10 ms
1 ms
22:45
22:50
p99
p95
med
avg

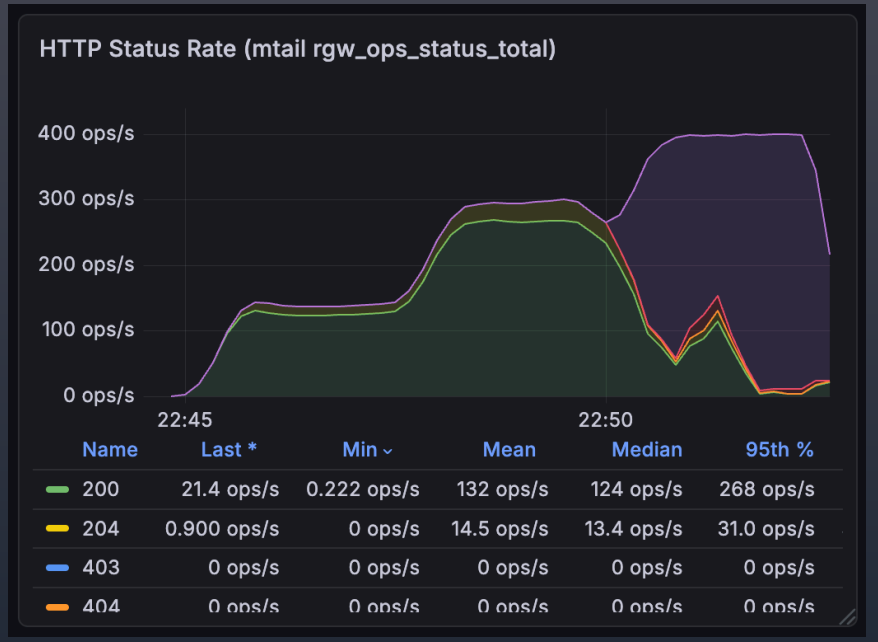
RGW Req In Flight

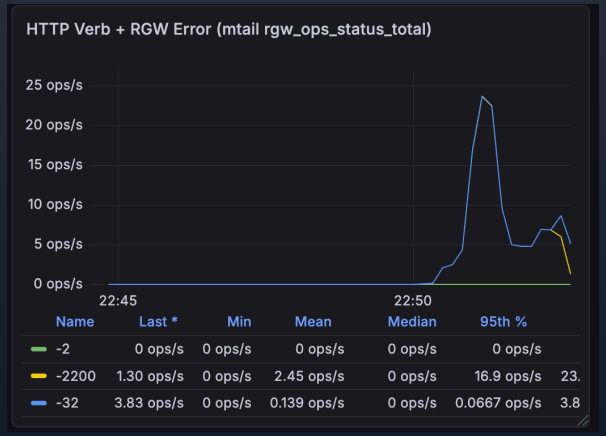


Latency Histogram All Requests




Req Rate
400 ops/s
300 ops/s
200 ops/s
100 ops/s
0 ops/s
22:45
22:50
reqs
fail


HTTP Status Rate (mtail rgw_ops_status_total)
400 ops/s
300 ops/s
200 ops/s
100 ops/s
0 ops/s
22:45
22:50
Name Last * Min Mean Median 95th %
200 21.4 ops/s 0.222 ops/s 132 ops/s 124 ops/s 268 ops/s
204 0.900 ops/s 0 ops/s 14.5 ops/s 13.4 ops/s 31.0 ops/s
403 0 ops/s 0 ops/s 0 ops/s 0 ops/s 0 ops/s
404 0 ops/s 0 ops/s 0 ops/s 0 ops/s 0 ops/s


HTTP Verb + RGW Error (mtail rgw_ops_status_total)
25 ops/s
20 ops/s
15 ops/s
10 ops/s
5 ops/s
0 ops/s
22:45
22:50
Name Last * Min Mean Median 95th %
-2 0 ops/s 0 ops/s 0 ops/s 0 ops/s 0 ops/s
-2200 1.30 ops/s 0 ops/s 2.45 ops/s 0 ops/s 16.9 ops/s 23.
-32 3.83 ops/s 0 ops/s 0.139 ops/s 0 ops/s 0.0667 ops/s 3.8

Active Benchmarking

Do it Live: Don’t wait for the report. There is real value in watching live dashboards as the system crosses from healthy, to constrained, to saturated.

Averages Hide the Truth.

Native RGW Histograms.

CL'SO

Powered by ceph

Thank you!

Marcel Lauhoff <marcel.lauhoff@clyso.com>

CL'SO

Bonus

CL'SO

mtail

```
histogram rgw_beast_latency_seconds buckets 0.005, 0.01, 0.025, 0.05, 0.1, 0.25, [...] by verb
histogram rgw_beast_content_length_bytes buckets 0, 1024, 2048, 4096, 16384, [...]
counter rgw_beast_ops_total by verb, http_status
counter rgw_beast_user_agents_total by verb, http_status, user_agent
counter rgw_beast_addr_total by verb, http_status, addr

/beast: \S+: (?P<addr>\S+) - \S+ .* \\?"(?P<verb>[A-Z]+) \S+ \S+\\?" (?P<http_status>\d+) (?
P<content_length>\d+) - \\?"(?P<user_agent>[^"\\]*)\\?" - latency=(?P<latency>\d+\.\d+)s/ {
  rgw_beast_latency_seconds[$verb] = float($latency)
  rgw_beast_content_length_bytes[$verb] = int($content_length)
  rgw_beast_ops_total[$verb][$http_status]++
  rgw_beast_user_agents_total[$verb][$http_status][$user_agent]++
  rgw_beast_addr_total[$verb][$http_status][$addr]++
}
```

k6

Run JavaScript
in a loop
with instrumentation
at scale

```
export default function () {
 http.get('https://ceph.com');
}
```

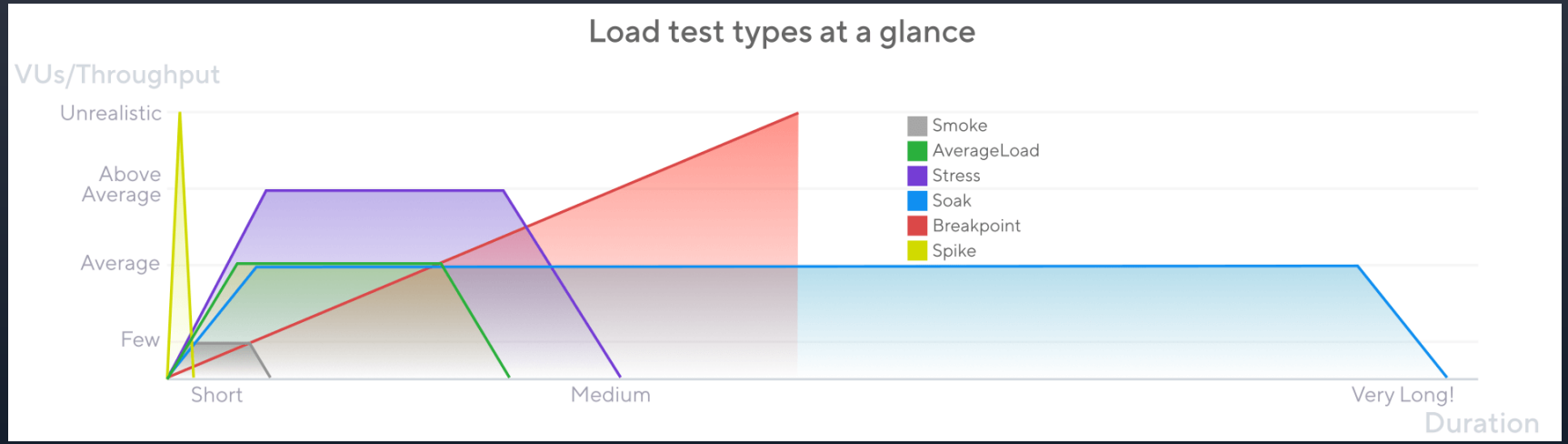
k6

Run JavaScript
in a loop
with instrumentation
at scale

https://github.com/grafana/k6-operator

k6

Run JavaScript
in a loop
with instrumentation
at scale



Source: https://grafana.com/docs/k6/latest/testing-guides/test-types/

CL'SO

k6

Run JavaScript
in a loop
with instrumentation
at scale

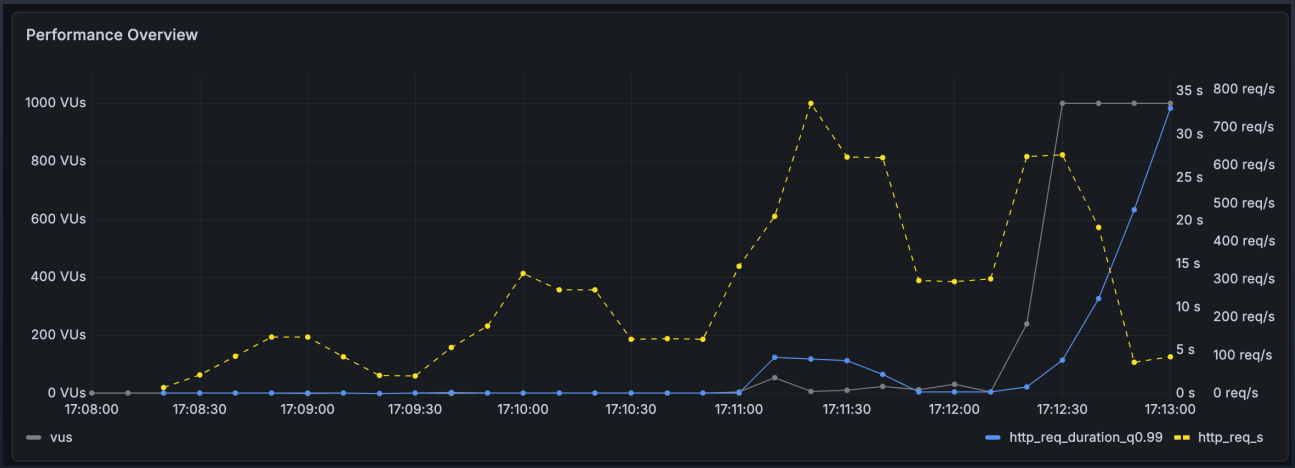
```
[...]
    checks_total.....: 90      13.122179/s
    checks_succeeded.....: 100.00% 90 out of 90
    checks_failed.....: 0.00%   0 out of 90

    HTTP
    http_req_duration.....: avg=140.36ms   min=119.08ms med=140.96ms max=154.63ms p(90)=146.88ms p(95)=148.21ms
      { expected_response:true }.....: avg=140.36ms   min=119.08ms med=140.96ms max=154.63ms p(90)=146.88ms p(95)=148.21ms
    http_req_failed.....: 0.00%  0 out of 45
    http_reqs.....: 45     6.56109/s
[...]
```

CL'SO

k6

Run JavaScript
in a loop
with instrumentation
at scale



Stream
metrics live to
Prometheus
and others