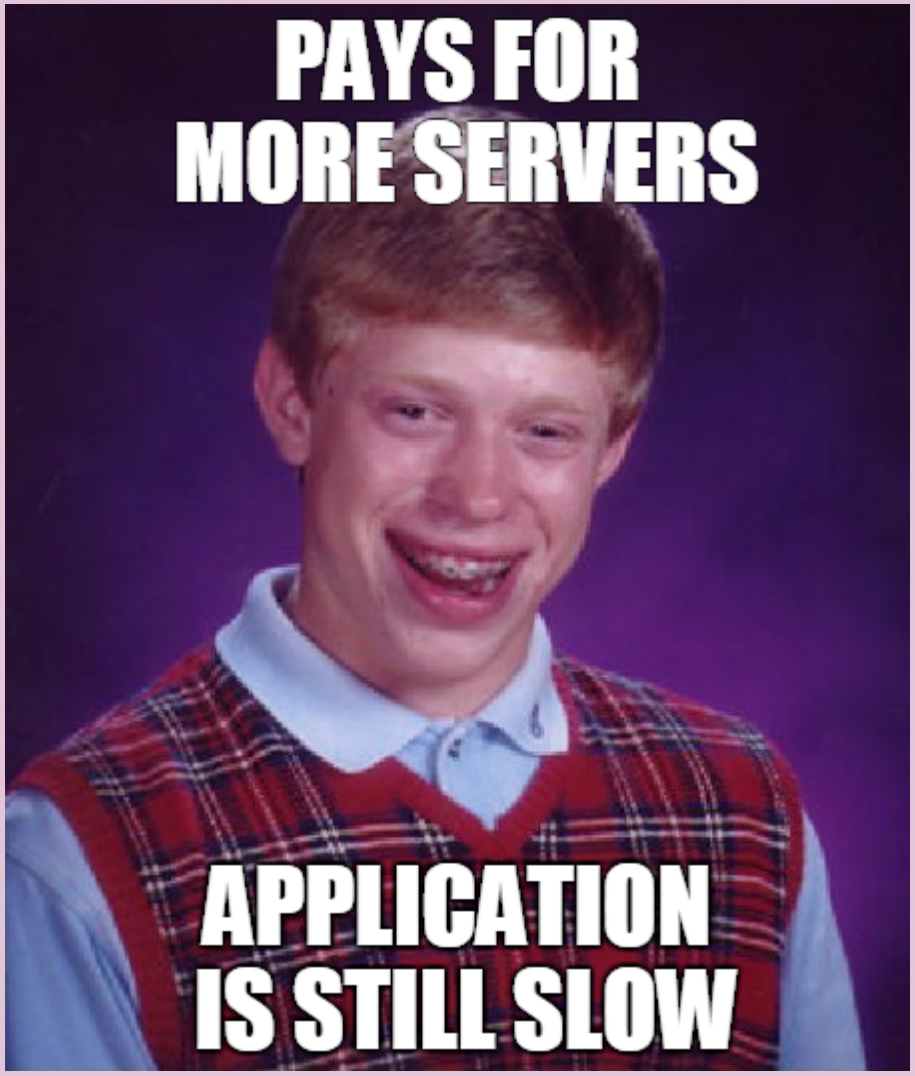
PERFORMANCE TESTING 101

MARIE CRUZ
Developer Advocate at k6.io
@mcruzdrake | testingwithmarie.com


212
Tip x 3
02:14


0
01:11

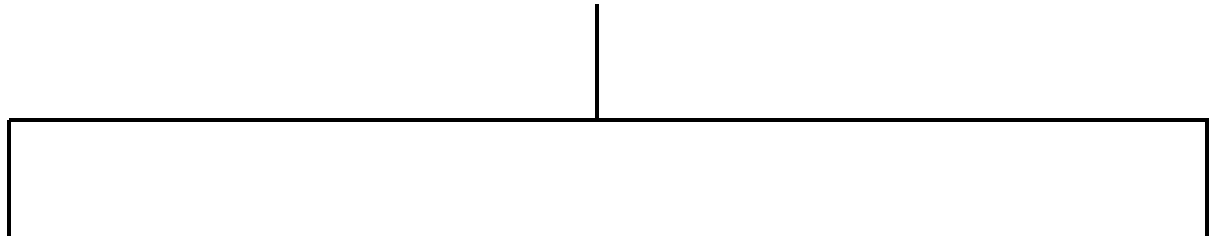

-10
-58
00:06


PAYS FOR
MORE SERVERS
APPLICATION
IS STILL SLOW

PERFORMANCE TESTING

PERFORMANCE TESTING

PERFORMANCE TESTING



PERFORMANCE TESTING

FRONT END/CLIENT SIDE

PERFORMANCE TESTING

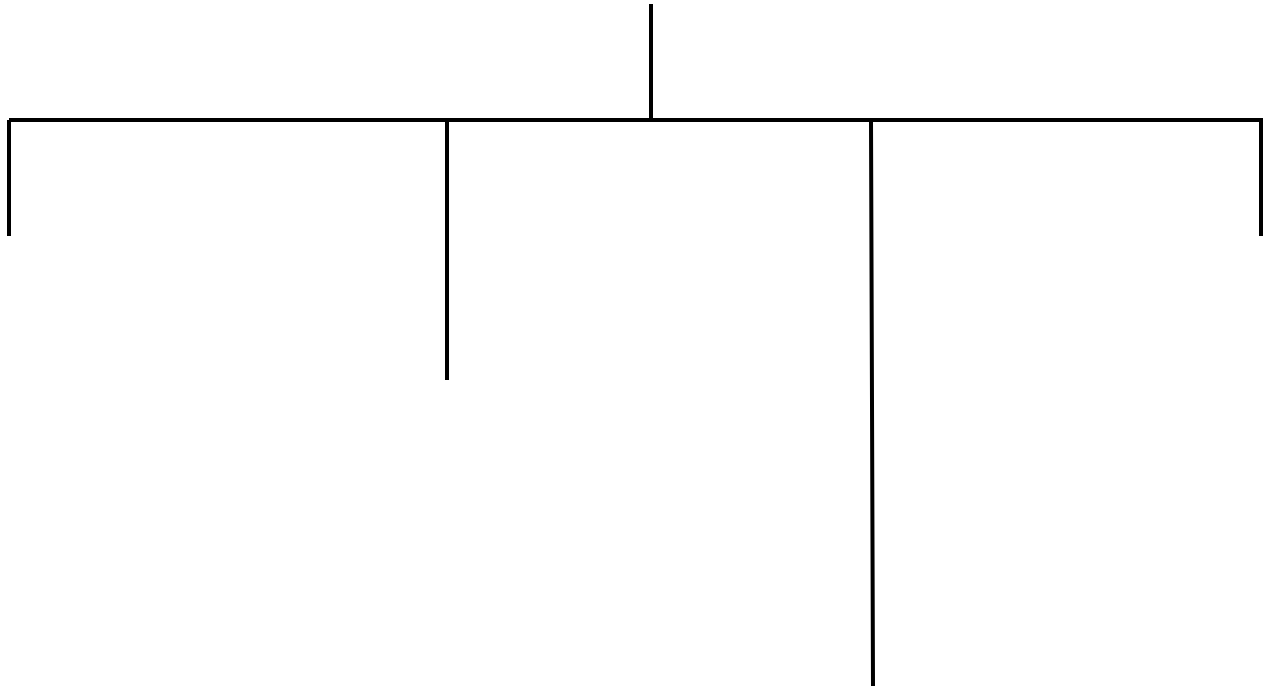
FRONT END/CLIENT SIDE

BACK END/SERVER SIDE

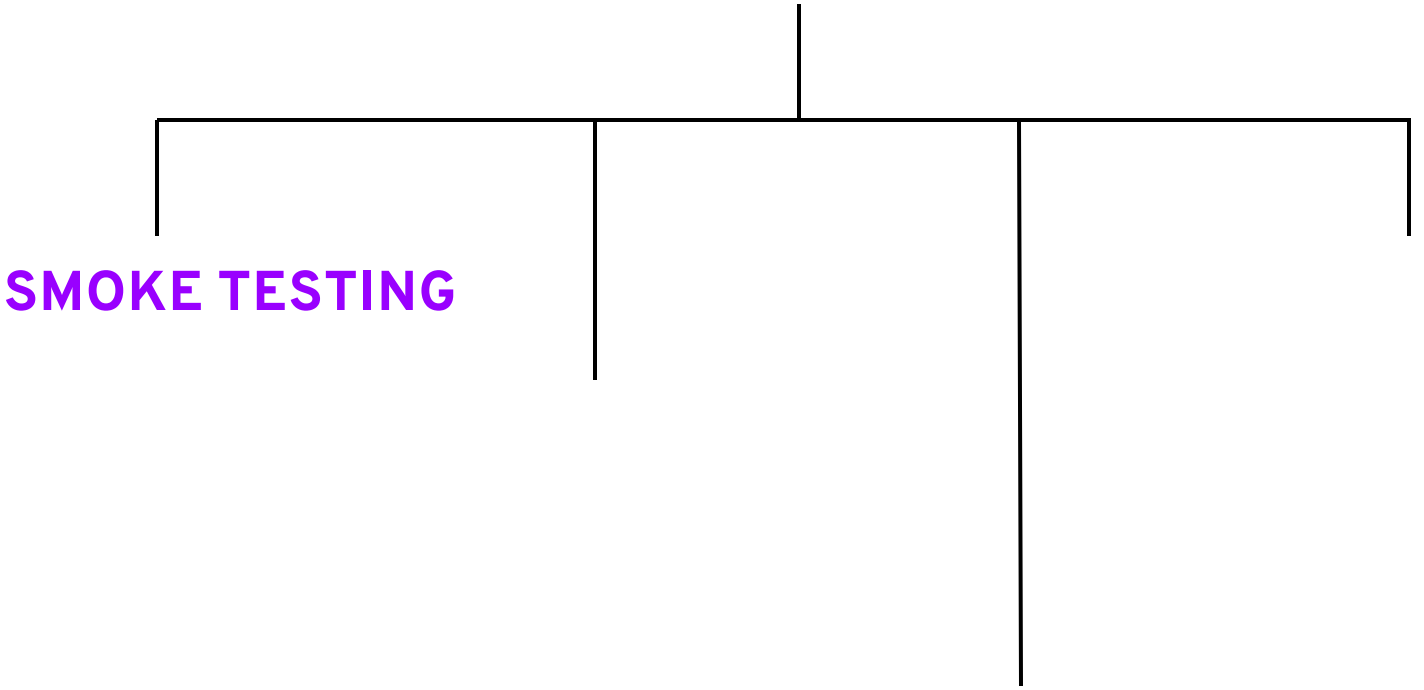
PERFORMANCE TESTING AND LOAD TESTING

LOAD TESTING

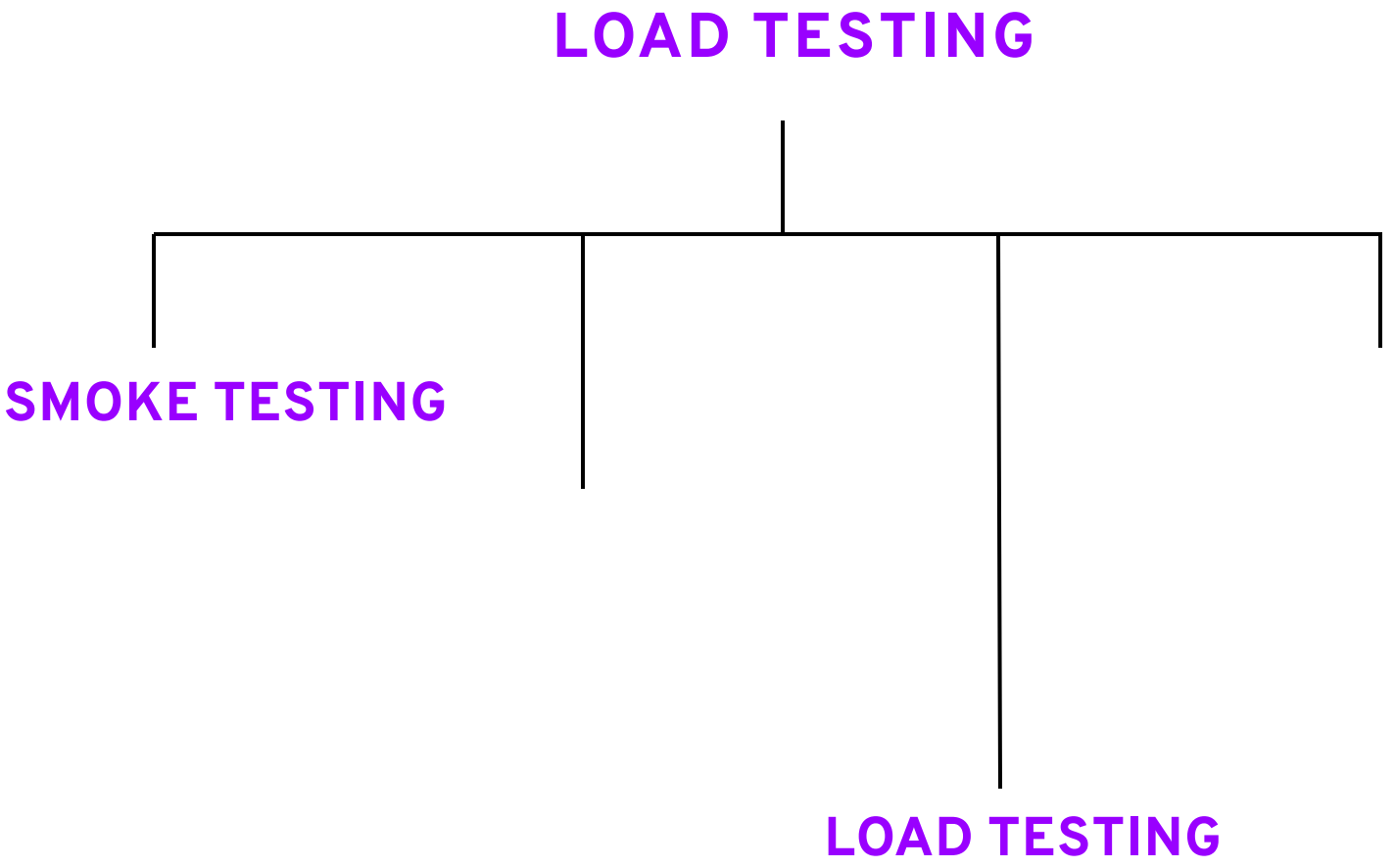
LOAD TESTING

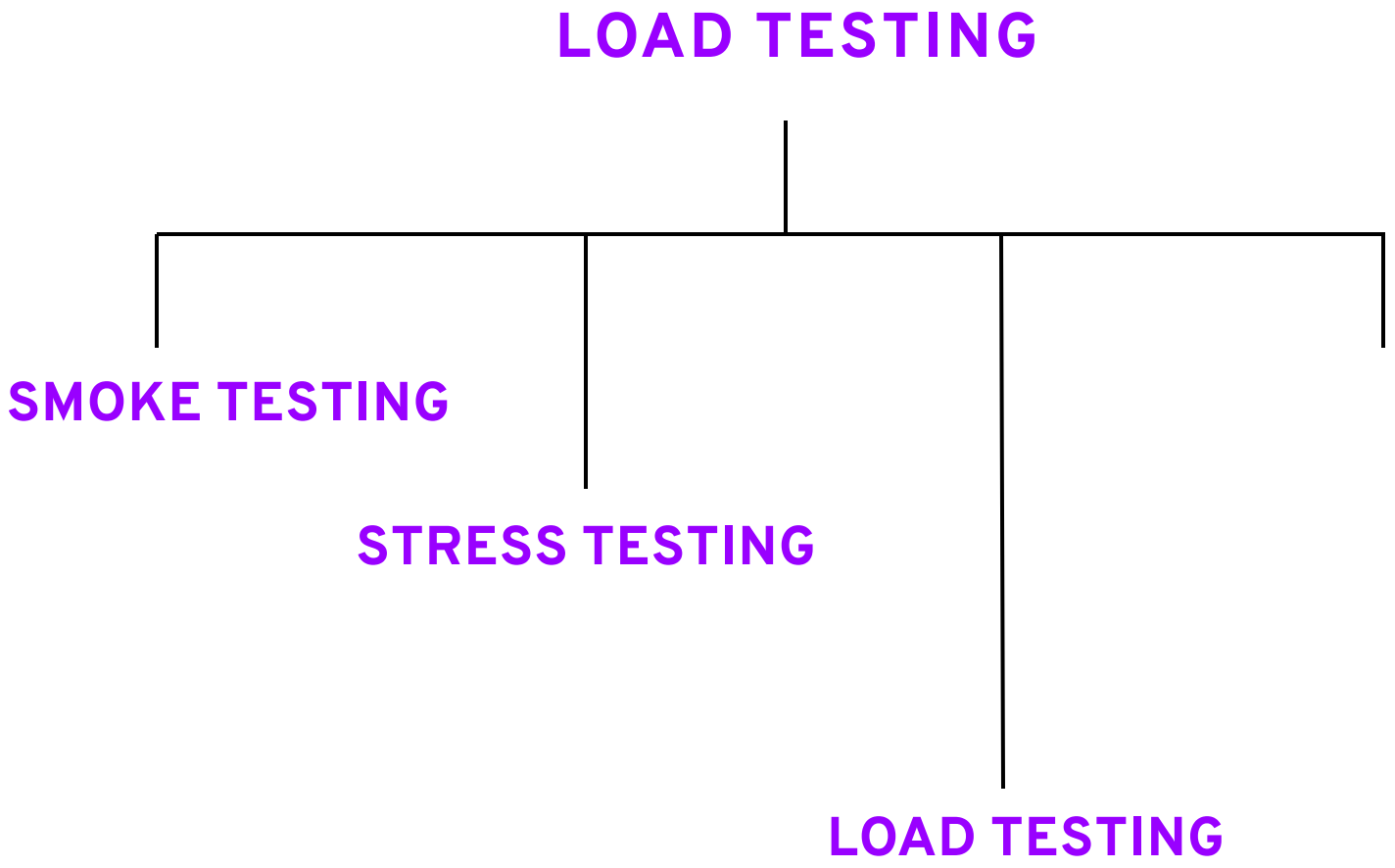


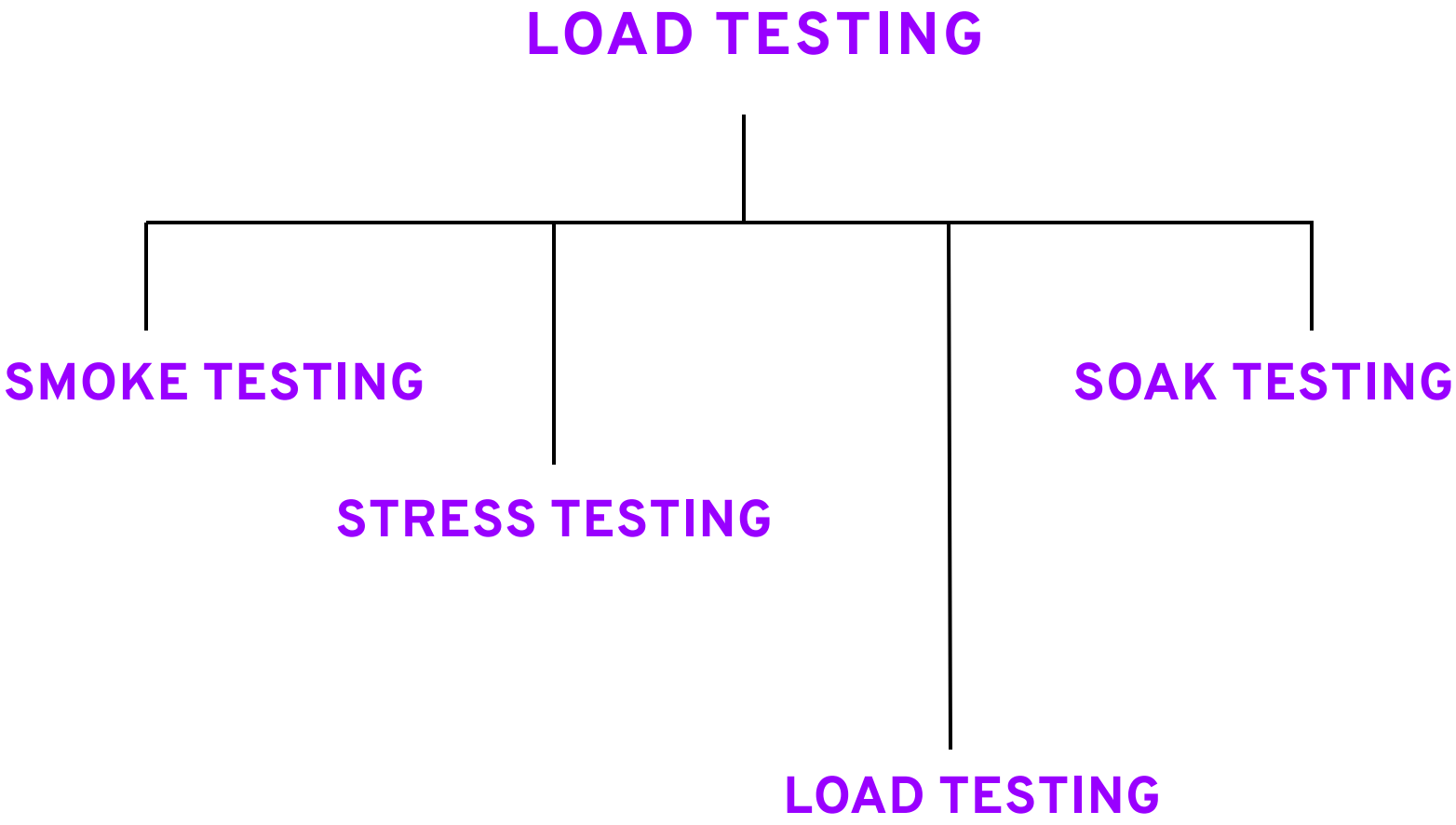
LOAD TESTING

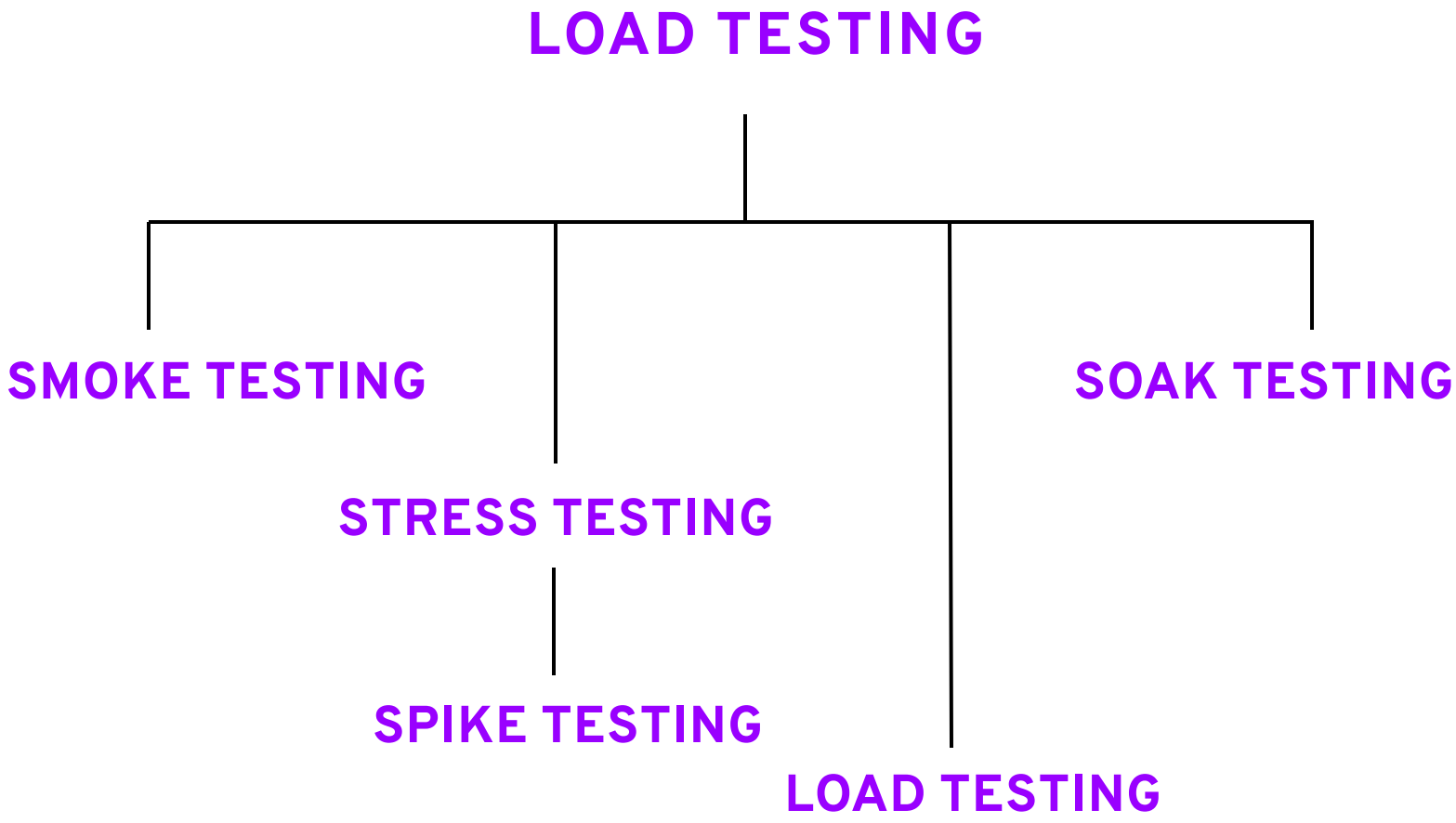


SMOKE TESTING

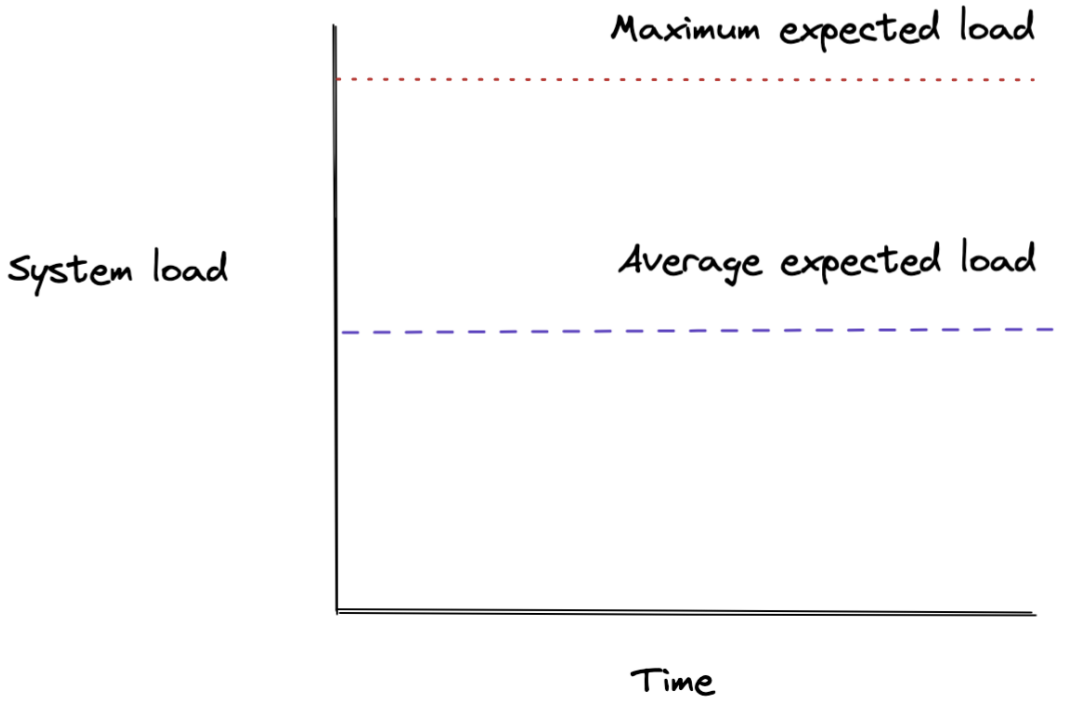

LOAD TESTING
SMOKE TESTING
LOAD TESTING


LOAD TESTING
SMOKE TESTING
STRESS TESTING
LOAD TESTING

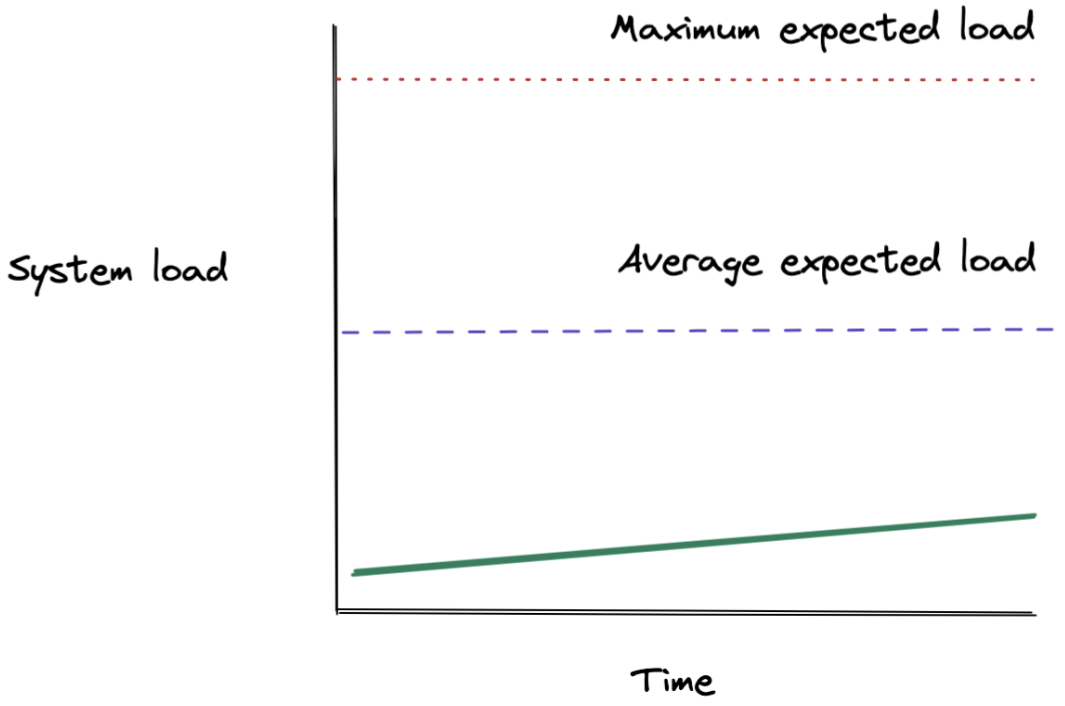

LOAD TESTING
SMOKE TESTING
STRESS TESTING
LOAD TESTING
SOAK TESTING


LOAD TESTING
SMOKE TESTING
STRESS TESTING
SPIKE TESTING
LOAD TESTING
SOAK TESTING

SMOKE TESTING

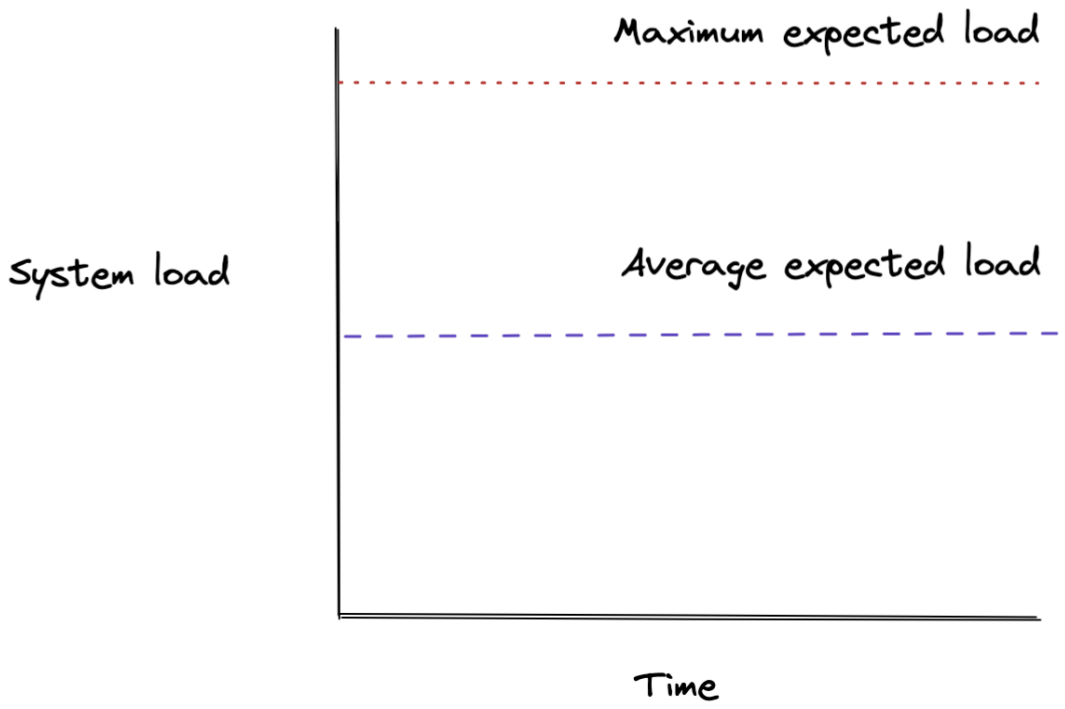


SMOKE TESTING

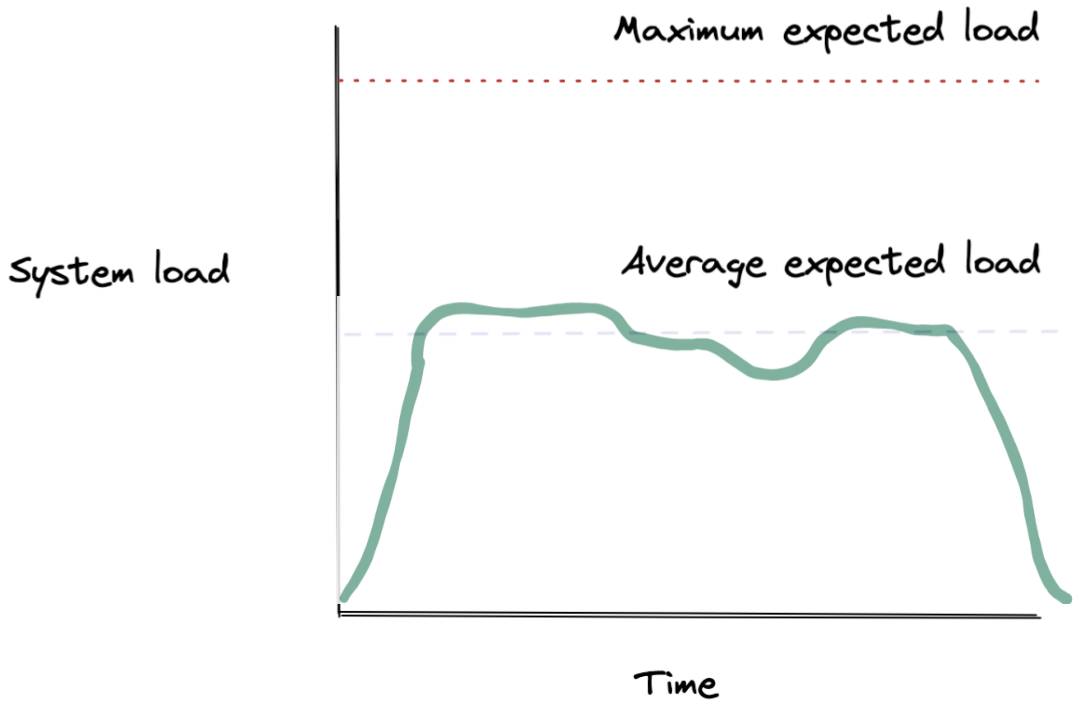


SMOKE TESTING

LOAD TESTING

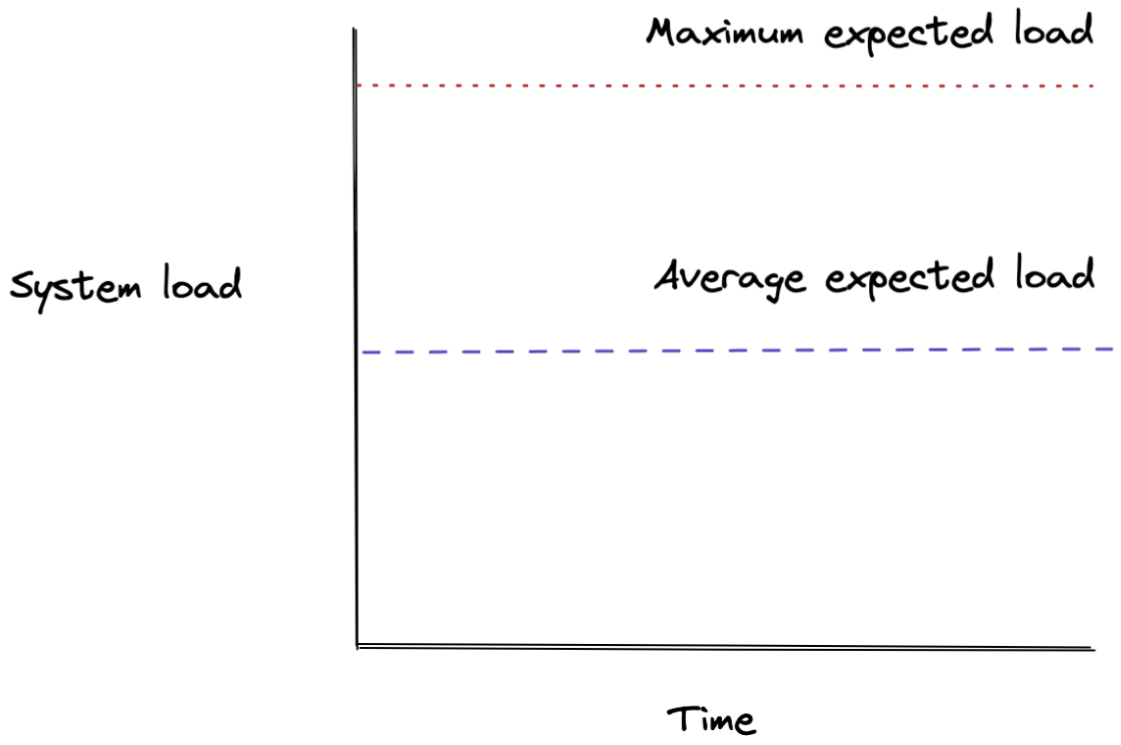


LOAD TESTING

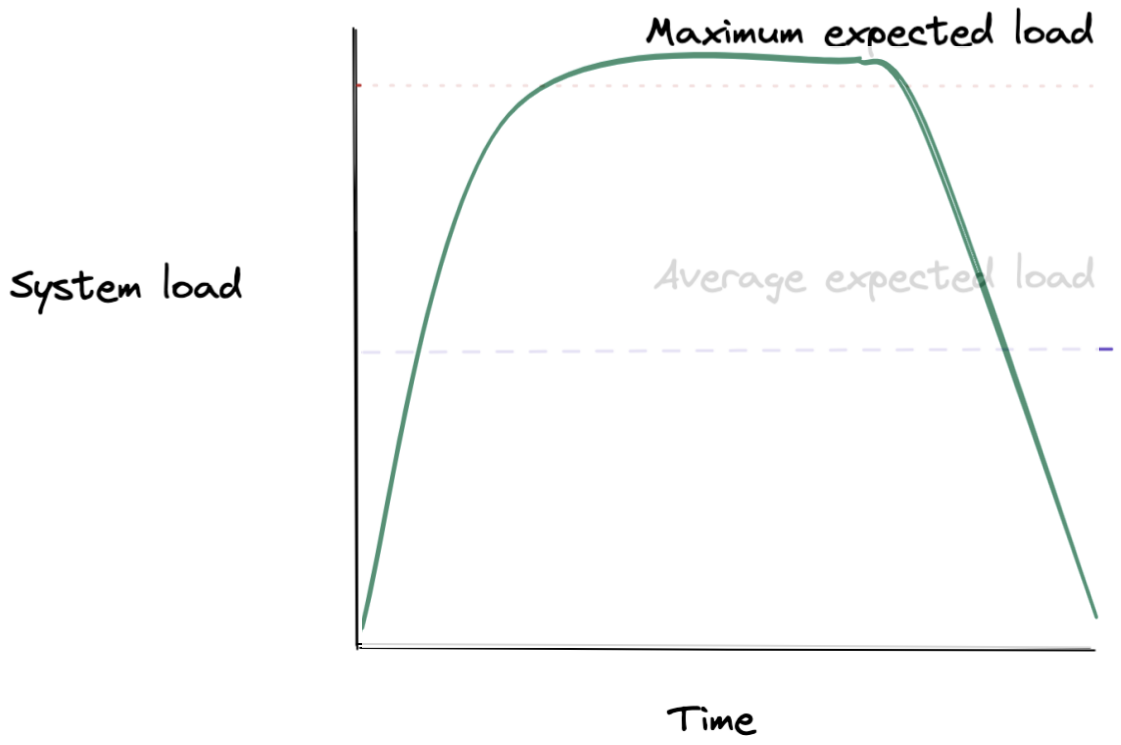


LOAD TESTING

STRESS TESTING

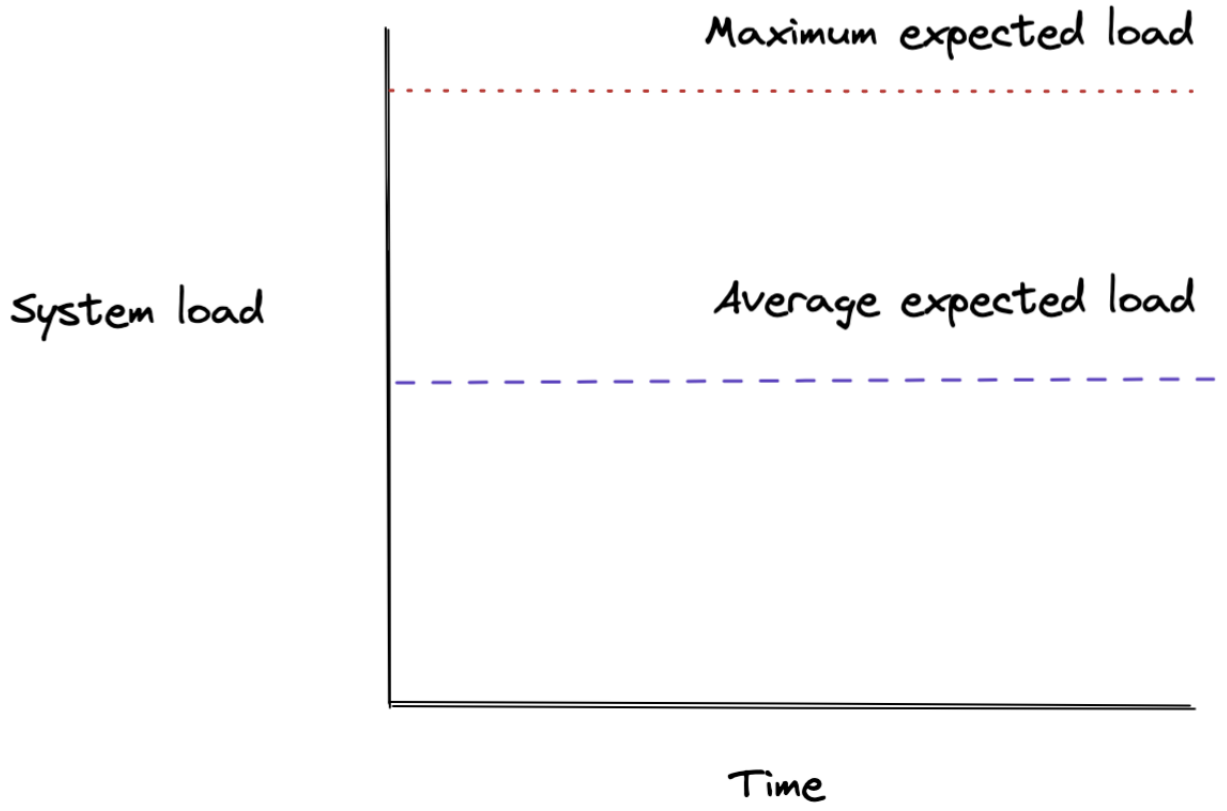


STRESS TESTING

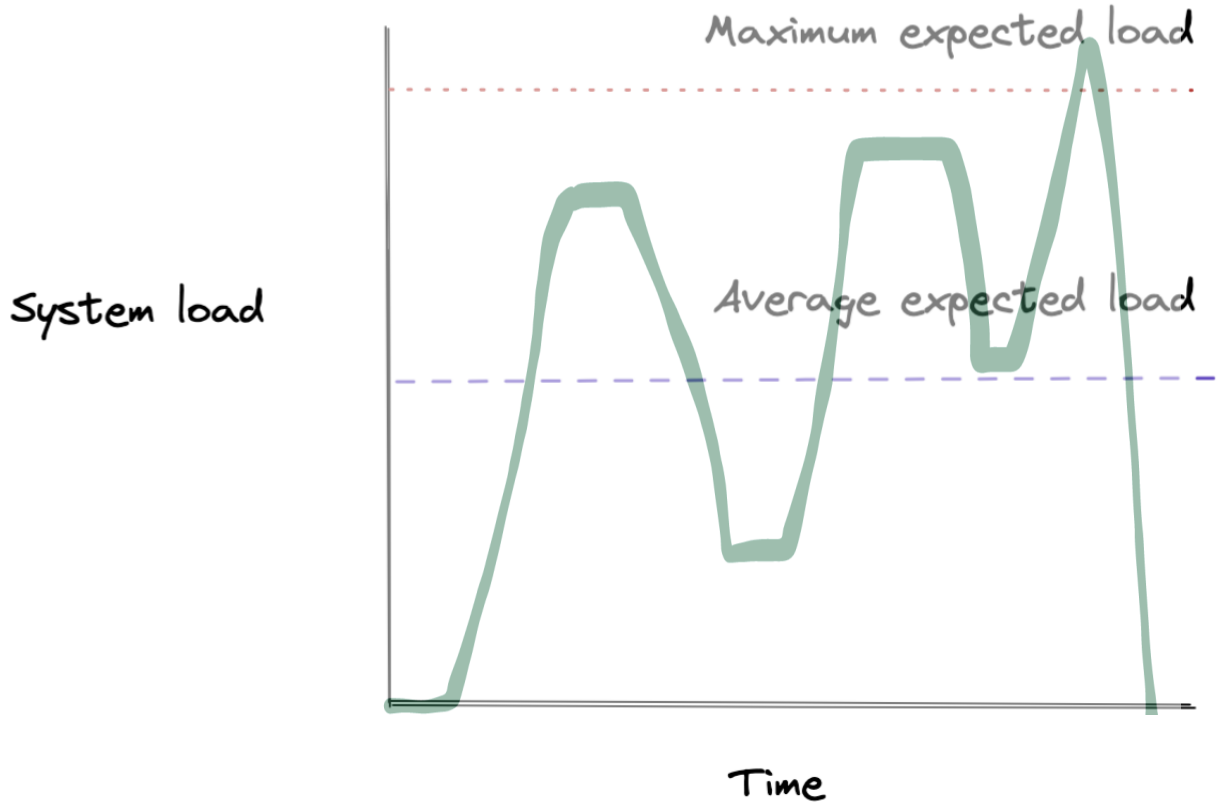


STRESS TESTING

SPIKE TESTING

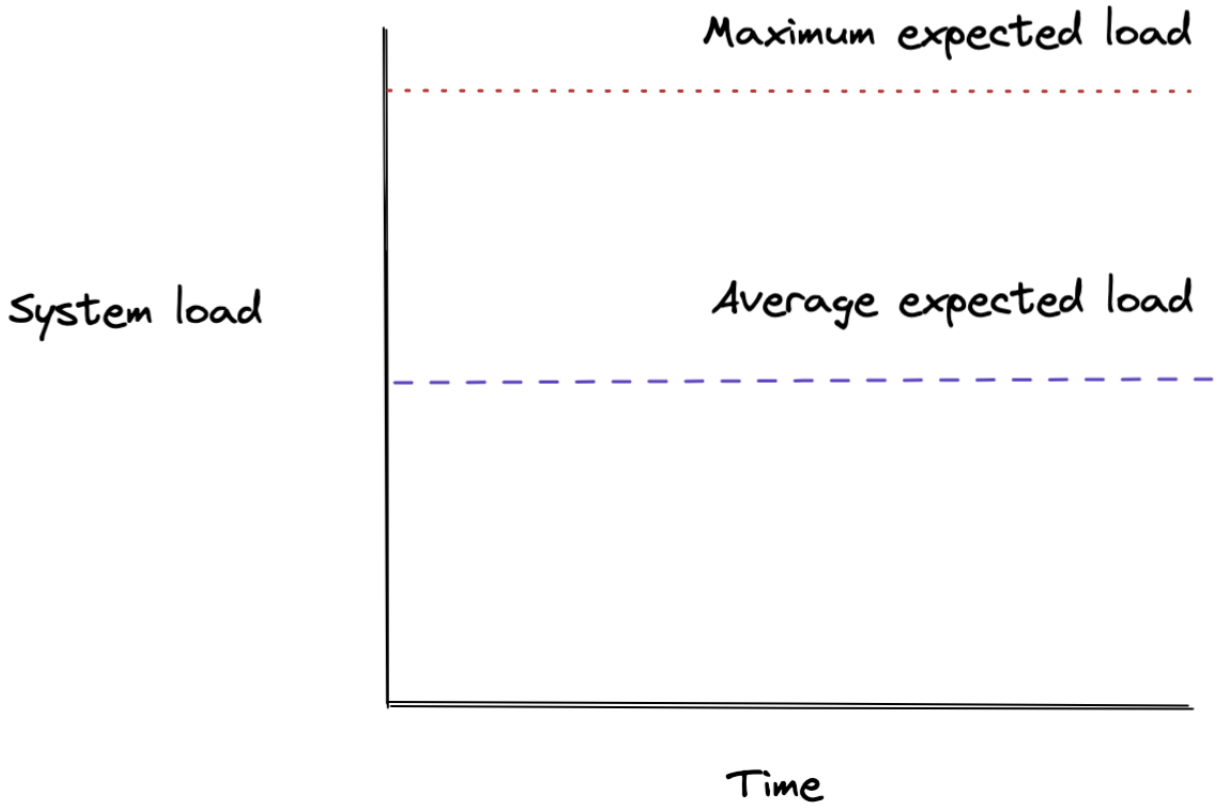


SPIKE TESTING

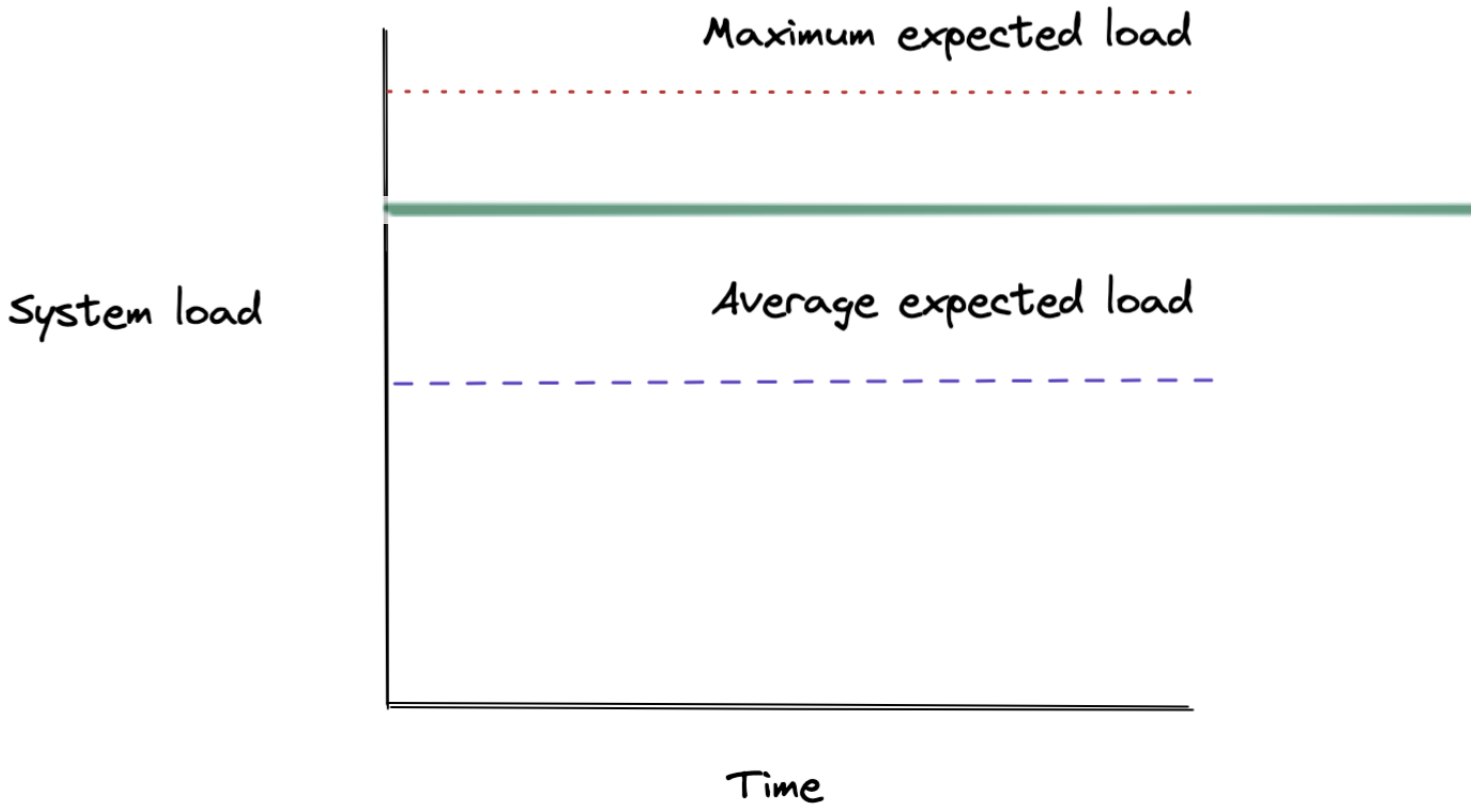


SPIKE TESTING

SOAK TESTING



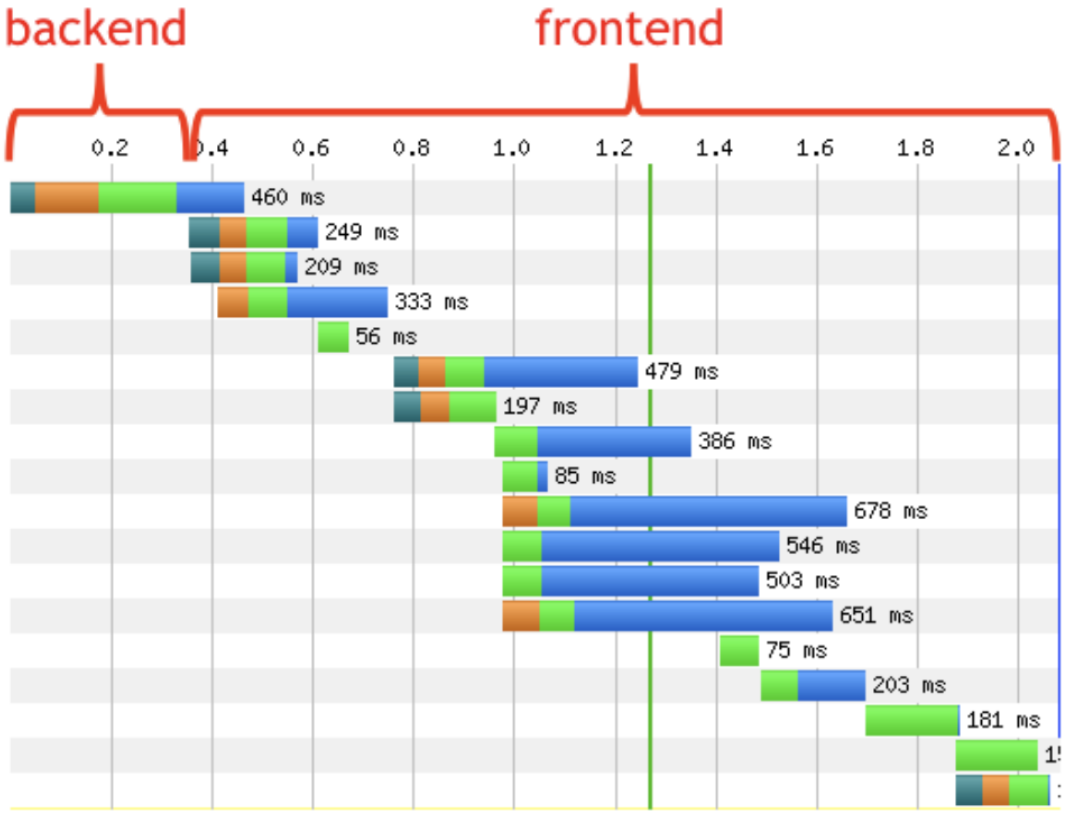
SOAK TESTING



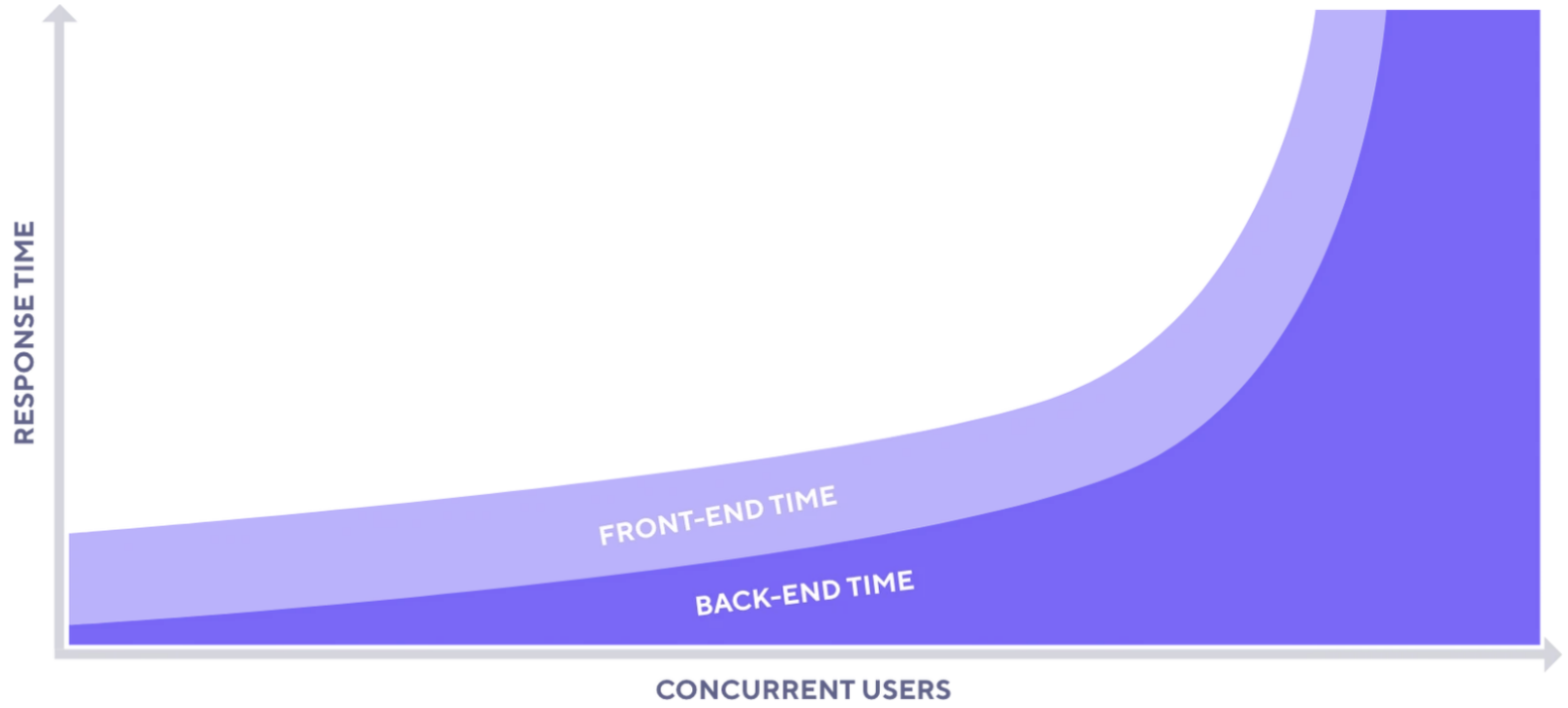
THE GOLDEN RULE OF WEB PERFORMANCE

THE GOLDEN RULE OF WEB PERFORMANCE

80-90% of the load time of a web page or application is spent in the frontend.



Resource: Steve Souders



Resource: k6.io

IT'S METRICS TIME!


If you can't measure it, you can't improve it" – Peter Drucker.
David Rothwell
quotefancy

PERFORMANCE TESTING METRICS

PERFORMANCE TESTING METRICS

CPU UTILISATION

PERFORMANCE TESTING METRICS

CPU UTILISATION

MEMORY UTILISATION

PERFORMANCE TESTING METRICS

CPU UTILISATION

MEMORY UTILISATION

LATENCY

PERFORMANCE TESTING METRICS

CPU UTILISATION

MEMORY UTILISATION

LATENCY

THROUGHPUT

PERFORMANCE TESTING METRICS

CPU UTILISATION

REQUEST TIME

MEMORY UTILISATION

LATENCY

THROUGHPUT

PERFORMANCE TESTING METRICS

CPU UTILISATION

REQUEST TIME

MEMORY UTILISATION

LATENCY

RESPONSE TIME

THROUGHPUT

PERFORMANCE TESTING METRICS

CPU UTILISATION

REQUEST TIME

MEMORY UTILISATION

LATENCY

FAILURE RATE

RESPONSE TIME

THROUGHPUT

PERFORMANCE TESTING METRICS

CPU UTILISATION

REQUEST TIME

MEMORY UTILISATION

LATENCY

FAILURE RATE

RESPONSE TIME

THROUGHPUT

NUMBER OF CONCURRENT VIRTUAL USERS (VUS)

K6

```
  data_received..................: 22 kB 5.7 kB/s
  data_sent......................: 742 B 198 B/s
  http_req_blocked...............: avg=1.05s    min=1.05s    med=1.05s    max=1.05s    p(90)=1.
  http_req_connecting............: avg=334.26ms min=334.26ms med=334.26ms max=334.26ms p(90)=33
  http_req_duration..............: avg=2.7s     min=2.7s     med=2.7s     max=2.7s     p(90)=2.
    { expected_response:true }...: avg=2.7s     min=2.7s     med=2.7s     max=2.7s     p(90)=2.
  http_req_failed................: 0.00% ✓ 0        ✗ 1
  http_req_receiving.............: avg=112.41µs min=112.41µs med=112.41µs max=112.41µs p(90)=11
  http_req_sending...............: avg=294.48µs min=294.48µs med=294.48µs max=294.48µs p(90)=29
  http_req_tls_handshaking.......: avg=700.6ms  min=700.6ms  med=700.6ms  max=700.6ms  p(90)=70
  http_req_waiting...............: avg=2.7s     min=2.7s     med=2.7s     max=2.7s     p(90)=2.
  http_reqs......................: 1     0.266167/s
  iteration_duration.............: avg=3.75s    min=3.75s    med=3.75s    max=3.75s    p(90)=3.
  iterations.....................: 1     0.266167/s
  vus............................: 1     min=1      max=1
  vus_max........................: 1     min=1      max=1
```

FRONT END/CLIENT SIDE

FRONT END/CLIENT SIDE

1. First Contentful Paint

FRONT END/CLIENT SIDE

1. First Contentful Paint
2. Large Contentful Paint

FRONT END/CLIENT SIDE

1. First Contentful Paint
2. Large Contentful Paint
3. Speed Index

FRONT END/CLIENT SIDE

1. First Contentful Paint
2. Large Contentful Paint
3. Speed Index
4. Time to Interactive

FRONT END/CLIENT SIDE

1. First Contentful Paint
2. Large Contentful Paint
3. Speed Index
4. Time to Interactive
5. Total Blocking Time

FRONT END/CLIENT SIDE

1. First Contentful Paint
2. Large Contentful Paint
3. Speed Index
4. Time to Interactive
5. Total Blocking Time
6. Cumulative Layout Shift

HOW TO GET STARTED?


keep it simple

hthouse report

Device

- ○ Mobile
- ◉ Desktop

Categories

- ☑ Performance
- ☐ Accessibility
- ☐ Best practices
- ☐ SEO
- ☐ Progressive Web App

Plugins

- ☐ Publisher Ads

Reports

12:06:44 - www.testingwithma

https://www.testingwithmarie.com/blog

92

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Collapse view

● First Contentful Paint

0.8 s

First Contentful Paint marks the time at which the first text or image is painted. Learn more.

■ Time to Interactive

2.7 s

Time to interactive is the amount of time it takes for the page to become fully interactive. Learn more.

■ Speed Index

1.3 s

Speed Index shows how quickly the contents of a page are visibly populated. Learn more.

● Total Blocking Time

10 ms

Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. Learn more.

■ Largest Contentful Paint

1.5 s

Largest Contentful Paint marks the time at which the largest text or image is painted. Learn more

● Cumulative Layout Shift

0.004

Cumulative Layout Shift measures the movement of visible elements within the viewport. Learn more.

View Original Trace | View Treemap

Show audits relevant to: All FCP TBT LCP CLS

OPPORTUNITIES

DIAGNOSTICS

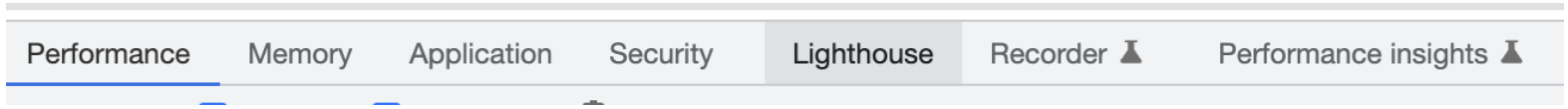
- Serve static assets with an efficient cache policy — 82 resources found
- Avoid chaining critical requests — 10 chains found
- User Timing marks and measures — 207 user timings
- Keep request counts low and transfer sizes small — 135 requests • 2,002 KiB
- Avoid large layout shifts — 5 elements found
- Avoid long main-thread tasks — 1 long task found

More information about the performance of your application. These numbers don't directly affect the Performance score.

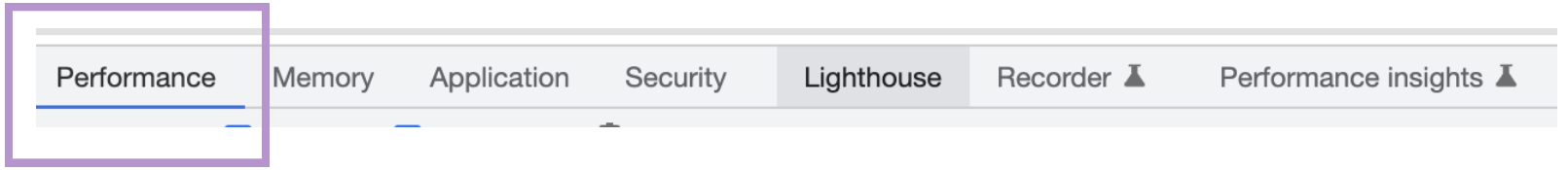
PASSED AUDITS (34) Hide

- Eliminate render-blocking resources
- Properly size images
- Defer offscreen images
- Minify CSS
- Minify JavaScript
- Reduce unused CSS — Potential savings of 14 KiB
- Reduce unused JavaScript — Potential savings of 91 KiB
- Efficiently encode images
- Serve images in next-gen formats
- Enable text compression
- Preconnect to required origins
- Initial server response time was short — Root document took 220 ms
- Avoid multiple page redirects
- Preload key requests
- Use HTTP/2

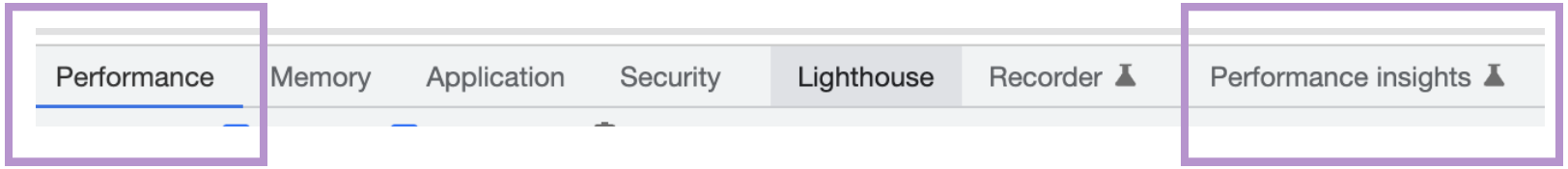
PERFORMANCE VS PERFORMANCE INSIGHTS TAB



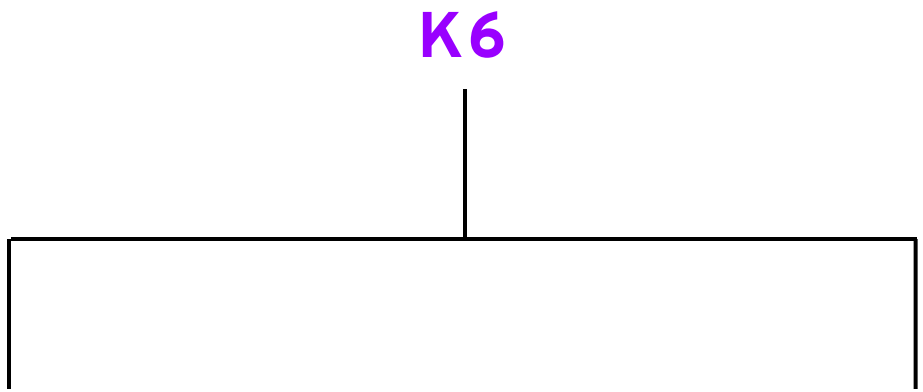
PERFORMANCE VS PERFORMANCE INSIGHTS TAB

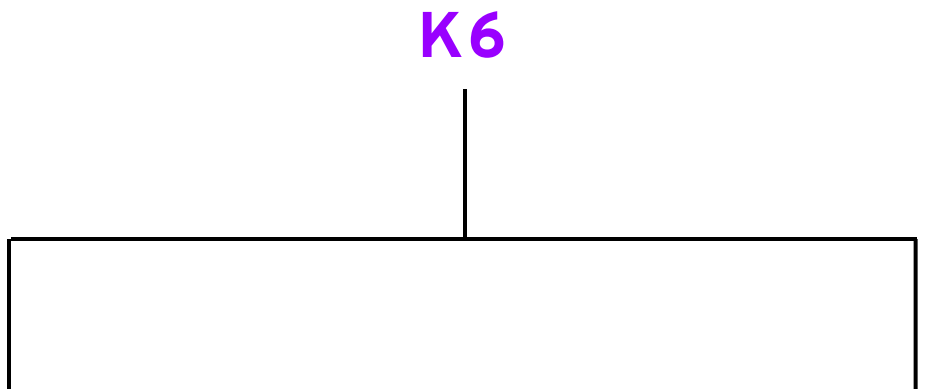


PERFORMANCE VS PERFORMANCE INSIGHTS TAB



K6

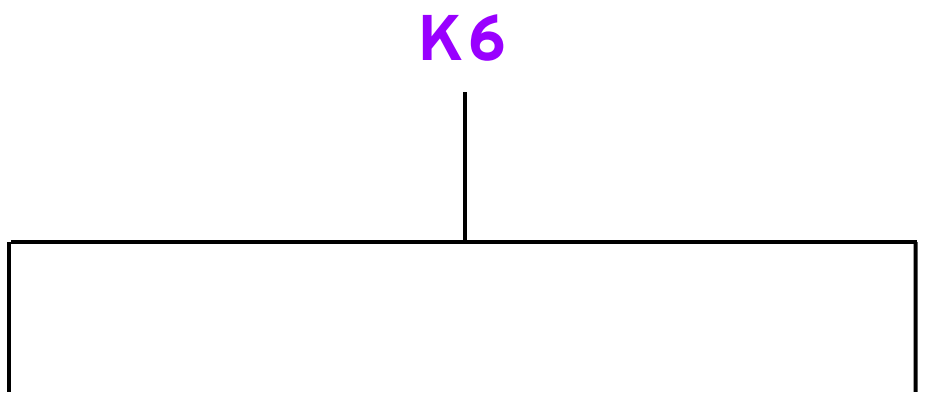

K6



k6 Open Source

A modern load testing tool built
for developer happiness

DOWNLOAD >_



k6 Open Source

A modern load testing tool built for developer happiness

DOWNLOAD >_

k6 Cloud

Managed performance testing for engineering teams

START FREE TRIAL >_

Schedule a Demo >

50 Free Cloud Tests

K6 AS CODE

```
1  import http from 'k6/http';
2  import { check, sleep } from 'k6';
3
4  export const options = {
5    stages: [
6      { duration: '30s', target: 20 },
7      { duration: '1m30s', target: 10 },
8      { duration: '20s', target: 0 },
9    ],
10 };
11
12 export default function () {
13   const res = http.get('https://httpbin.test.k6.io/');
14   check(res, { 'status was 200': (r) => r.status == 200 });
15   sleep(1);
16 }
```

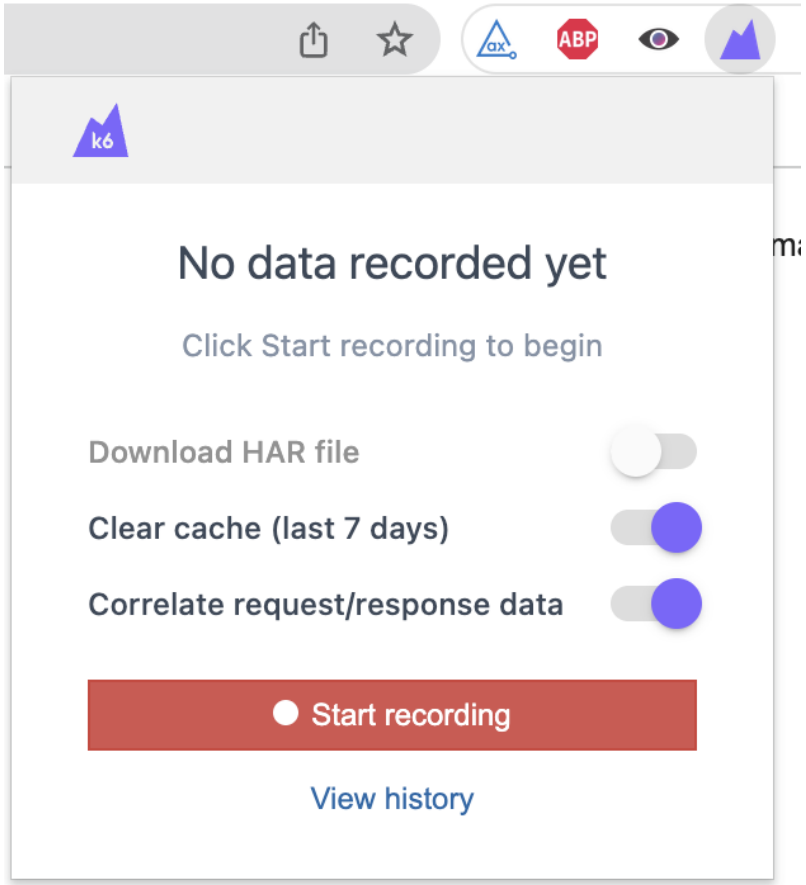
K6 AS CODE

```
import http from 'k6/http';
import { check, sleep } from 'k6';

export const options = {
  stages: [
    { duration: '30s', target: 20 },
    { duration: '1m30s', target: 10 },
    { duration: '20s', target: 0 },
  ],
};

export default function () {
  const res = http.get('https://httpbin.test.k6.io/');
  check(res, { 'status was 200': (r) => r.status == 200 });
  sleep(1);
}
```

K6 AS CODE

```
1  import http from 'k6/http';
2  import { check, sleep } from 'k6';
3
4  export const options = {
5    stages: [
6      { duration: '30s', target: 20 },
7      { duration: '1m30s', target: 10 },
8      { duration: '20s', target: 0 },
9    ],
10 };
11
12 export default function () {
13   const res = http.get('https://httpbin.test.k6.io/');
14   check(res, { 'status was 200': (r) => r.status == 200 });
15   sleep(1);
16 }
```

K6 BROWSER RECORDER



SAVE YOUR RECORDED TEST

Select project and a test name, check the filtered third-party and static assets settings and choose a test type you want to save as

My first project | Recorded test (01/09/2022-14:02:05)

Test builder

Use our code samples as a foundation for your script or start from a clean slate.

Script editor

Use our interactive UI to compose GET, POST, PUT and PATCH requests.

Correlate request and response data

Automatically detect and set up variables for data returned from the server.

Include static assets

Select to include **1** static assets requests (fonts, images, css, js etc).

Generate sleep

Automatically generate `sleep()` between requests that are made ≥ 500ms apart.

Third-party domains filtering

We have found and disabled **113** requests made to **5** different **third-party domains**. Select the ones you want to include in the test.

SELECT ALL

- frog.wix.com (17 requests)
- static.parastorage.com (54 requests)
- siteassets.parastorage.com (6 requests)
- fonts.gstatic.com (6 requests)
- static.wixstatic.com (36 requests)

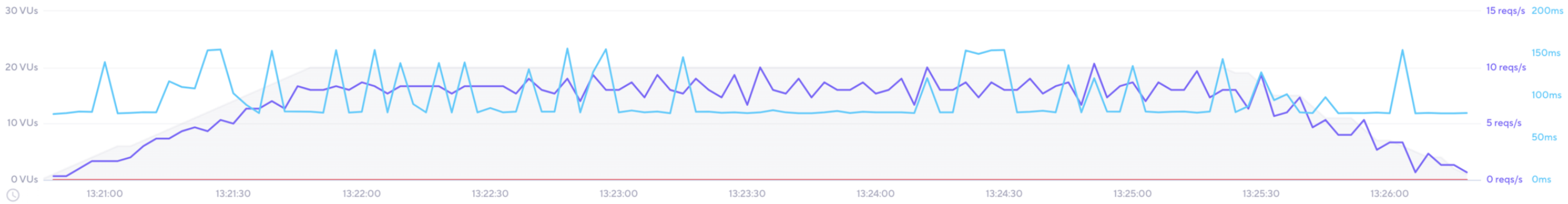
SAVE

FINALIZING TEST 20 VUs 6min Load zones: London Started by: marie.drake@grafana.com

PERFORMANCE OVERVIEW

The test is running...

REQUESTS MADE	HTTP FAILURES	PEAK RPS	P95 RESPONSE TIME
2 250 reqs	0 reqs	10.33 reqs/s	86 ms



SCENARIOS (1) HIDE SHOW

PERFORMANCE INSIGHTS

Analyzing test data...

THRESHOLDS (2/2) | CHECKS (1.1K/1.1K) | HTTP (2.3K/2.3K) | ANALYSIS Compare metrics | SCRIPT View executed script | LOGS Execution logs

FILTERS Filter expression...

NAME	
✓ http_req_failed: rate<10	-
✓ http_req_duration: p(95)≤500	p(95)=86

SUMMARY

1. **FRONTEND AND BACKEND PERFORMANCE TESTING ARE EQUALLY AS IMPORTANT.**

1. **FRONTEND AND BACKEND PERFORMANCE TESTING ARE EQUALLY AS IMPORTANT.**
2. **PERFORMANCE TESTING IS NOT JUST ABOUT LOAD TESTING.**

1. FRONTEND AND BACKEND PERFORMANCE TESTING ARE EQUALLY AS IMPORTANT.
2. PERFORMANCE TESTING IS NOT JUST ABOUT LOAD TESTING.
3. IF YOU CAN'T MEASURE IT, YOU CAN'T IMPROVE IT.

1. FRONTEND AND BACKEND PERFORMANCE TESTING ARE EQUALLY AS IMPORTANT.
2. PERFORMANCE TESTING IS NOT JUST ABOUT LOAD TESTING.
3. IF YOU CAN'T MEASURE IT, YOU CAN'T IMPROVE IT.
4. LEVERAGE TOOLS SUCH AS LIGHTHOUSE FOR FRONTEND AND K6 FOR BACKEND.


0
01:11

PERFORMANCE TESTING 101

MARIE CRUZ
Developer Advocate at k6.io
@mcruzdrake | testingwithmarie.com