

Why only measuring
uptime isn't enough to meet
user-expectations

PATIENTS

CAREGIVERS

RESILIENCE

DIGITAL ONCOLOGY

Living better with cancer every day

The app that supports cancer patients on their path to
resilience.



Download in
the App Store



Available on
Google Play

Pablo Seminario
@pabluk



Dr. Maria-Alice Franzoi

Medical oncologist and
researcher at Gustave Roussy

WHAT

“The ratio of the amount of time that a service was up”

$$MTTF / (MTTR + MTTF)$$

MTTF (Mean time to failure): which means how long the service was running before there was an outage

MTTR (Mean time to repair): how long it took during an outage to resolve

WHY

- Customer contracts
- SLAs
- Compliance
- Reporting



HOW

- Blackbox checks
- Pingdom
- Status Page

History

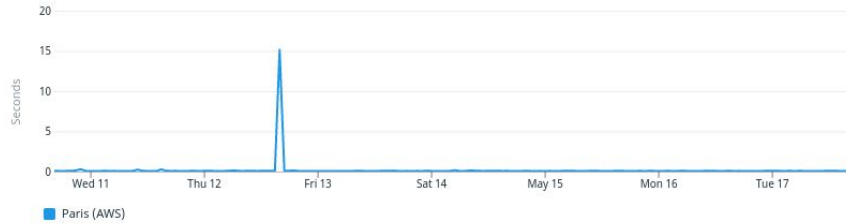
Show All Locations

Global Uptime

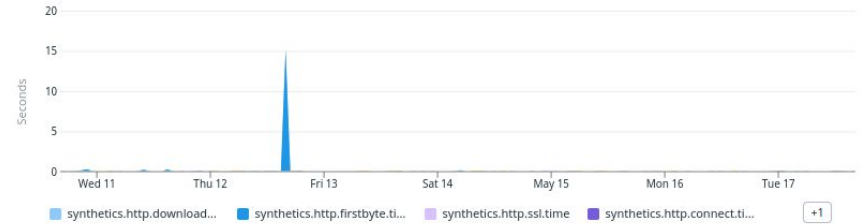
99.95%



Response Time by Location



Network timings (averaged)



Test Runs

Events

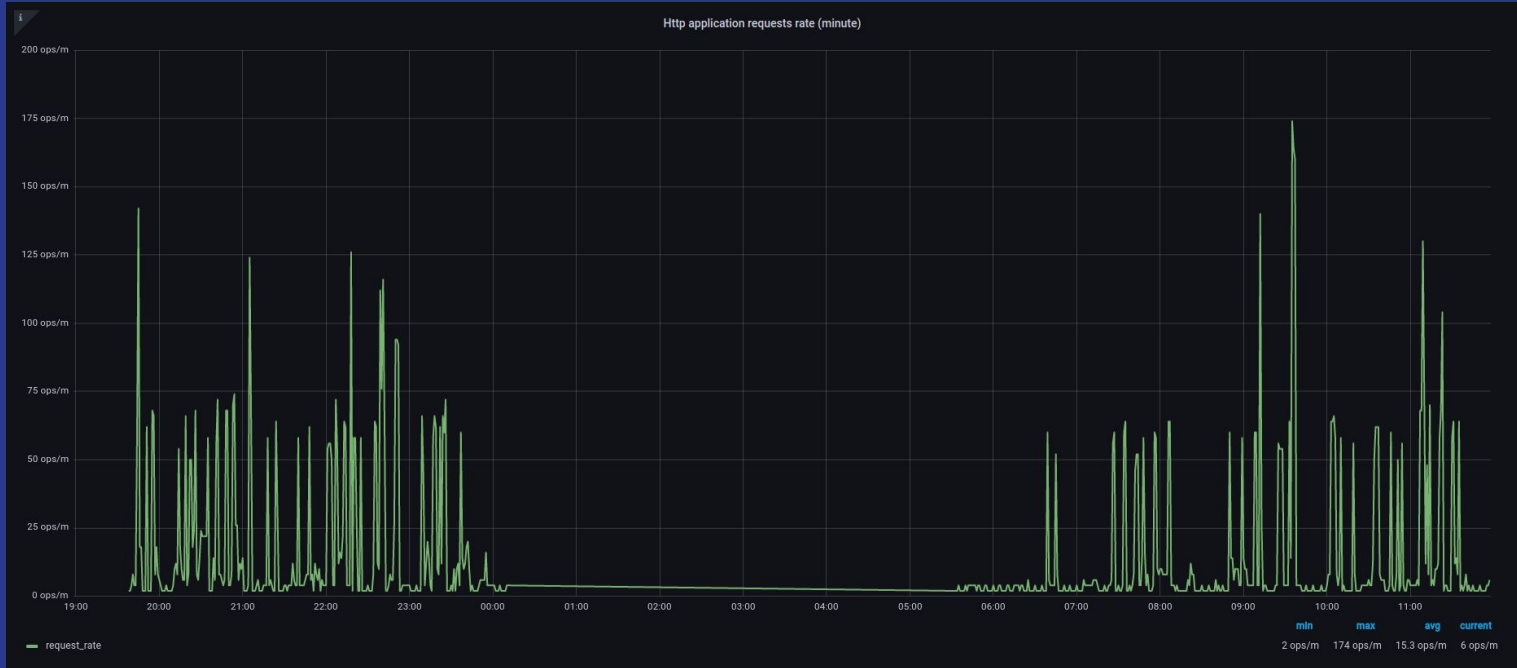
Test Runs

All Passed Failed

Showing the last 150 test runs between May 10, 4:23 pm and May 17, 4:23 pm for selected locations — Limit reached, refine your query if needed

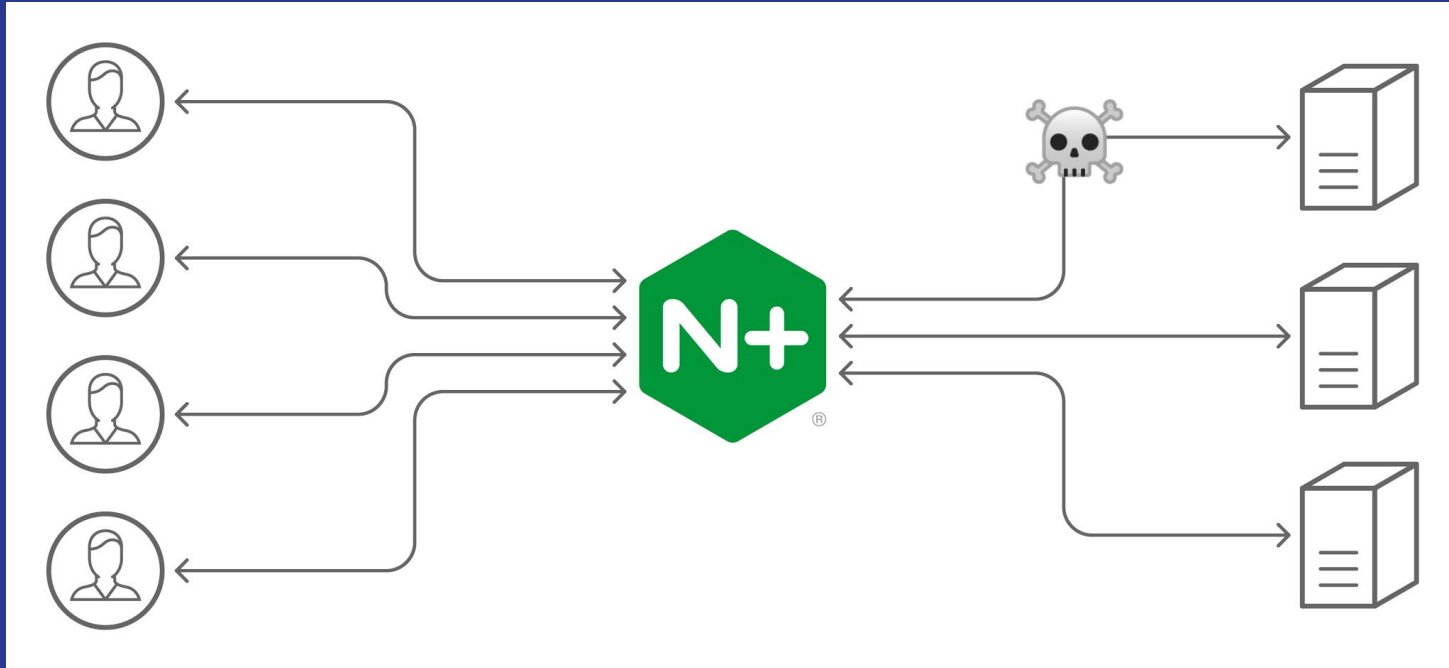
STATUS	DATE	LOCATION	RUN TYPE
PASSED	1 week ago May 17, 2022, 4:20 pm	Paris (AWS)	Scheduled
PASSED	1 week ago May 17, 2022, 4:15 pm	Paris (AWS)	Scheduled

But... is it still useful when



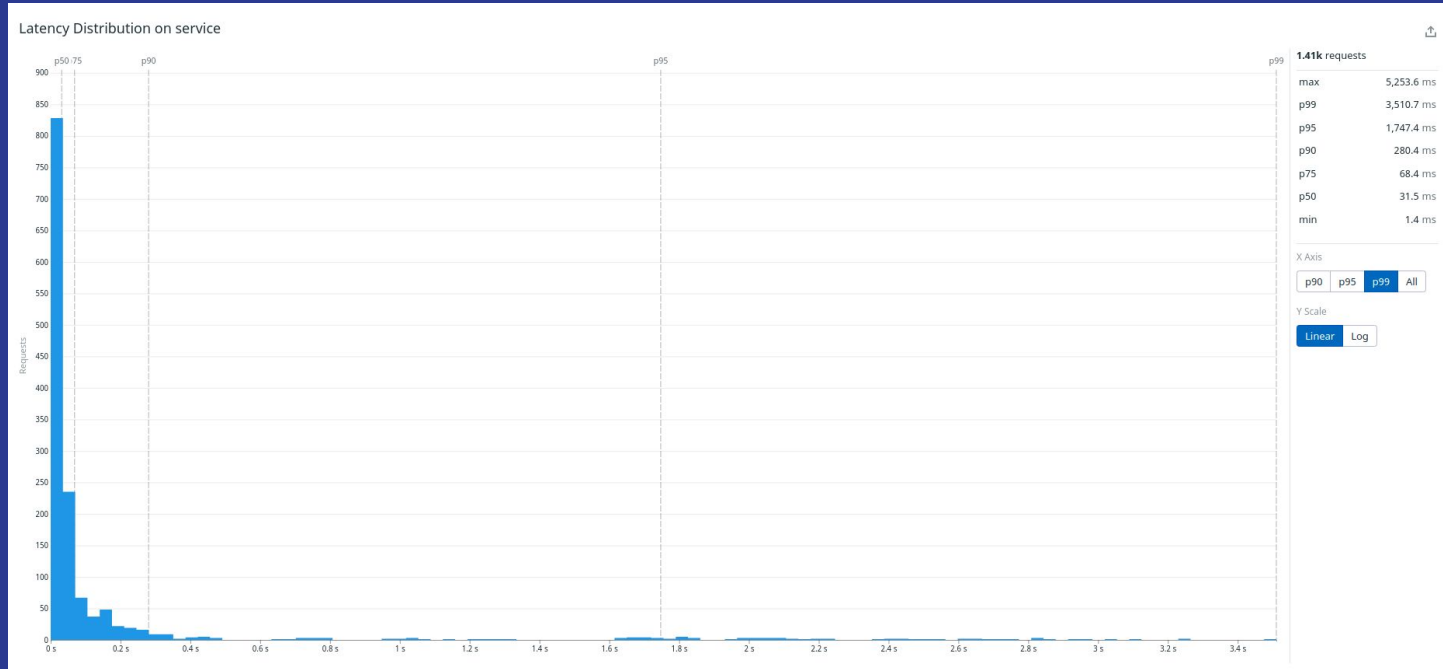
failures happen with no active users?

But... is it still useful when



only when server within a cluster is

But... is it still useful when



response time isn't the same for every



Golden signals

Latency

Traffic

Errors

Saturation

Count-based Availability Metrics

Success Rate = Good Events / Total Events

(Total HTTP Requests - HTTP Errors Requests) / Total HTTP Requests

```
http_requests_total{host="api", status="200"} / http_requests_total{host="api", status="200"}
```

```
(sum:trace.koa.request.hits{service:api}.as_count()-sum:trace.koa.request.errors{service:api}.as_count()) /  
(sum:trace.koa.request.hits{service:api}.as_count())
```

SLOs

@ SLOs > Edit SLO

Editing Availability SLO for `service`

1 Define the source

Good events (numerator)

a `trace.koa.request.hits` from `env: , service:` sum by (everything) as count Σ </>

b `trace.koa.request.errors` from `env: , service:` sum by (everything) as count Σ </>

\rightarrow a - b Σ </>

+ Add Query + Add Formula

Total events (denominator)

a `trace.koa.request.hits` from `env: , service:` sum by (everything) as count Σ </>

+ Add Query + Add Formula

2 Set your targets

Select up to 3 targets. Thresholds must be more than 0% and less than 100%. [?](#)

Target: % Time Window: Warning: %

+ New Target

3 Add name and tags

Name:

[Edit](#) [Preview](#) H B I S

This SLO tracks the availability of the `service`. Availability is measured as the number of successful requests divided by the number of total requests for the service

Get started with SLOs

With Service Level Objectives (SLOs) you can set one or more targets to the performance of a service. **Monitor Based** indicators use the uptime of one or more monitors as a source. **Metric Based** indicators are calculated by dividing the number of good events by the total number of events. [More info](#)

	STATUS	ERROR BUDGET
Past 30 Days:	100.000%	100% (0.9 reqs)

“My biggest learning is that the SLO mindset is definitely a marathon, so be prepared for it and pace yourself!”

Ioannis Georgoulas

<https://geototti21.medium.com/slo-from-nothing-to-production-91b8d4270bd5>

References

Notes: Meaningful Availability

<https://squidarth.com/systems/2021/01/08/meaningful-availability.html>

Available . . . or not? That is the question

<https://cloud.google.com/blog/products/gcp/available-or-not-that-is-the-question-cre-life-lessons>

The Four Golden Signals

https://sre.google/sre-book/monitoring-distributed-systems/#xref_monitoring_golden-signals

Implementing SLOs

<https://sre.google/workbook/implementing-slos/>

SLO — From Nothing to... Production

<https://geototti21.medium.com/slo-from-nothing-to-production-91b8d4270bd5>



Thanks!