Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Enabling performance insights: generating telemetry from software services developed at INFN-CNAF

Jacopo Gasparetto (INFN–CNAF)

Workshop sul calcolo nell'INFN | La Biodola 26-30 Maggio 2025

Outline

- Introduction to OpenTelemetry
- Telemetry data: Traces, Metrics and Logs
- Instrumentation
- Collecting telemetry data
- Case study: Otello
- Examples at INFN-CNAF
- Conclusions

Introduction to telemetry: OpenTelemetry

From the official documentation

> *OpenTelemetry is:*
> - ***An observability framework** and toolkit designed **to create** and manage telemetry data such as **traces**, **metrics**, and **logs**.*
> - ***Not an observability backend** like Jaeger, Prometheus, or other commercial vendors.*
> - *…*

OpenTelemetry is a collection of components that provide an SDK (Software Development Kit) to enrich an existing code base to *emit* telemetry data.

OpenTelemetry **does not** store any data *per se* and some kind of database(s) is(are) needed

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Telemetry data: Traces, Metrics and Logs

Telemetry data is generally grouped in three main entities, or *signals*: traces, metrics and logs.

- **traces**: represent **the path of a request through an application**. They are made up by *spans* that represent the single units of work or operation. Each span has **start and end timestamps**, a **name**, a **parent id** (if child of another span) and **attributes**. A typical trace/span information answers "*how much time this function took to execute?*" and "*what is the runtime call stack at this endpoint?*"
- **metrics**: represent the **measurement emitted by a meter**. Meters can typically be monotonic/non-monotonic counters, histograms and gauges. In the physical world "3 kWh at 2025-02-10 12:00" is the metric produced by the electricity meter (monotonic) of our house. An example could be: "how many requests we received so far at each endpoint?"
- **logs**: time-stamped text record, either structured or unstructured, with optional metadata. We do not deal with telemetry logs, as they could easily be written with a simple access/error logger

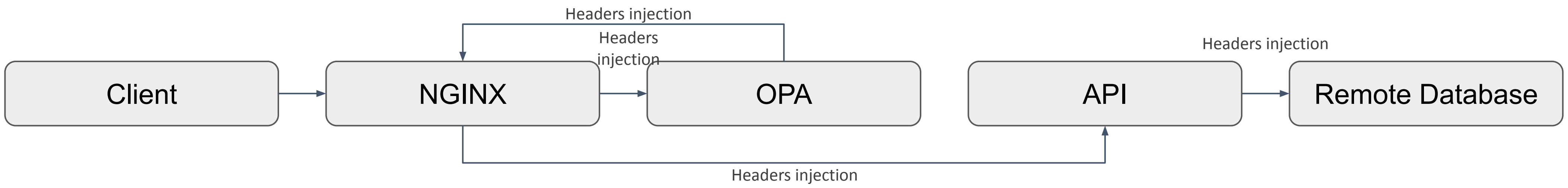
OpenTelemetry: Instrumentation

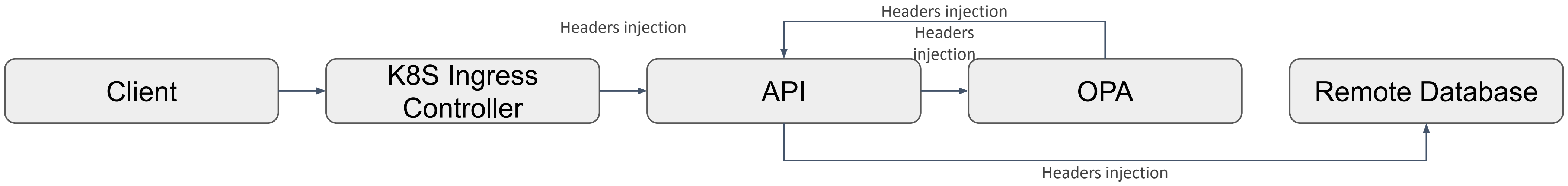
- The process of *augmenting* a codebase to emit telemetry data is called **instrumentation**. This means that we are adding code to our existing code.
- OpenTelemetry has SDKs for basically every language on the market.
- For some high level languages, such has JavaScript/TypeScript, OpenTelemetry offers the so called Zero-code Instrumentation feature that enables telemetry with just a couple of lines of configuration.
- For other languages such as C++, there is no automatic instrumentation and the codebase must be manually instrumented at each function of interest and it is up to the developer to choose what they want to monitor.
- A combined approach of manual and automatic instrumentation is always possible

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Propagation

Context propagation is performed by injecting headers by a service in the incoming HTTP request forwarded to the next service





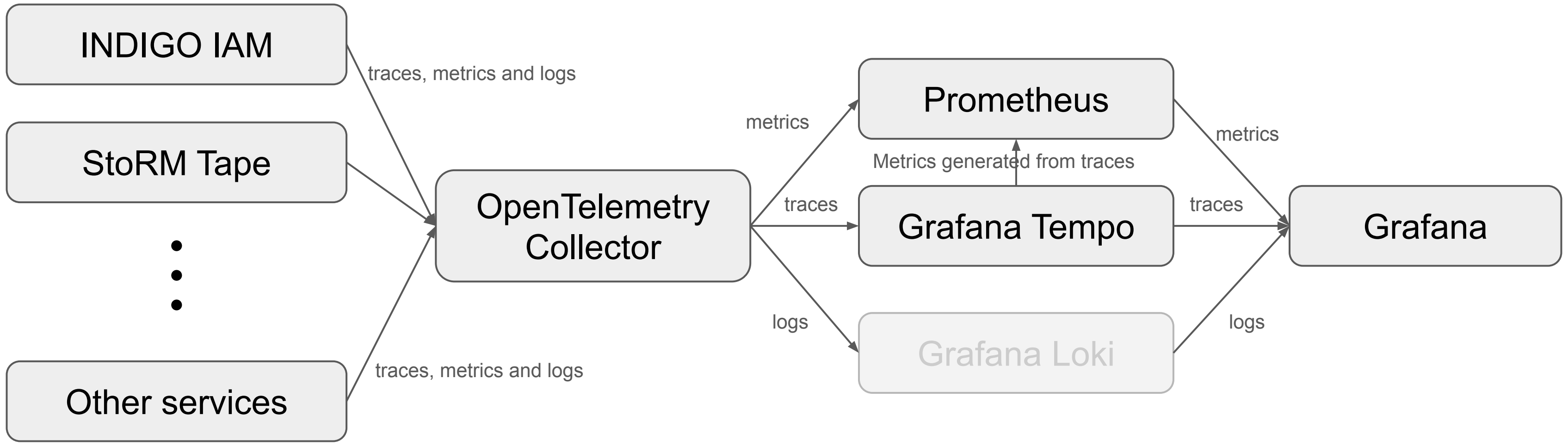
Response paths not shown in figure

Collect telemetry data

- An application can be composed of several components (micro-services), each sending telemetry data from a different endpoint but still logically representing the same application. For example, a trace with its root span can be opened by a NGINX reverse proxy that proxies the request to an API. It is important to reconstruct those traces to understand the complete path of the request.
- An application can have manyfold replicas, each sending telemetry data.
- Due to the nature of the data, different signals (traces, metrics and logs) must be stored in different kind of databases. They can also be sent through different protocols (HTTP, gRPC, kafka, ect.).
- OpenTelemetry offers a service called OpenTelemetry Collector which receives, processes and then exports telemetry data to the appropriate backends.

Otello: collecting telemetry data case study

- For our case study, we choose Prometheus as metrics backend and Grafana Tempo as traces backend
- Grafana is then used as web application to visualize KPIs, stats and plots
- A particular useful feature of Grafana Tempo is its built-in metrics generator which produces new metrics from the traces, such as the total count of requests and the latency histogram
- Otello is the prototype stack of services developed at INFN-CNAF used to collect telemetry data produced by software developed by the Software Development team



Otello: collecting telemetry data case study

Currently it is implemented as a docker compose composed by the following services:

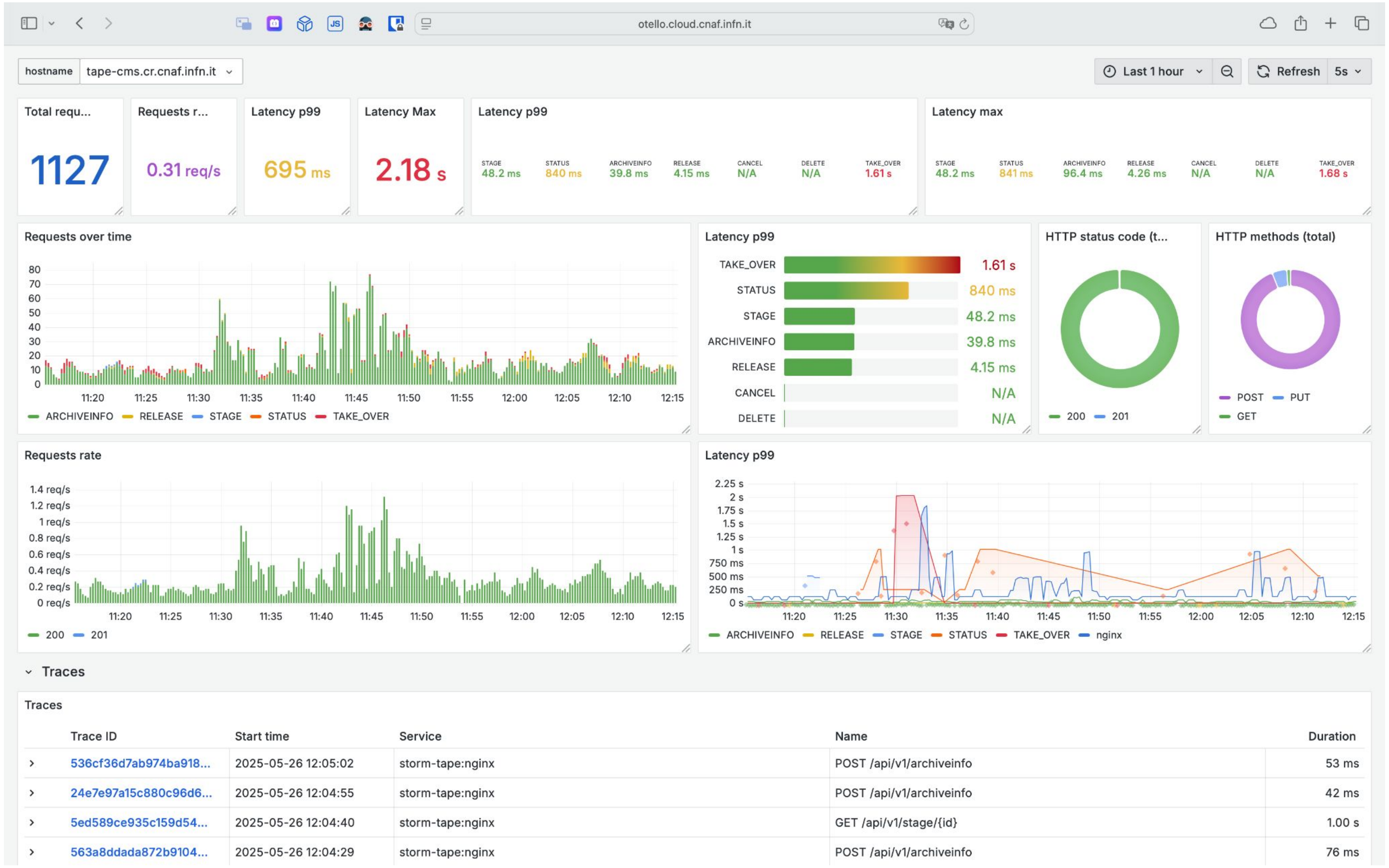
- NGINX (reverse proxy)
- OpenTelemetry Collector (HTTP + gRPC)
- Grafana Tempo (traces database)
- Prometheus (metrics database)
- Grafana (data visualization) with custom dashboards

Website: https://otello.cloud.cnaf.infn.it (available only inside CNAF network)
Respository: https://baltig.infn.it/cnafsd/opentelemetry

Instrumented software at INFN-CNAF

- StoRM Tape (C++)
- StoRM WebDAV (Java, testing)
- INDIGO IAM (Java, testing)
- New IAM Dashboard (Javascript)
- Data Cloud S3 webapp (Javascript)

StoRM Tape



hostname: tape-cms.cr.cnaf.infn.it

Last 1 hour · Refresh 5s

Total requ...	Requests r...	Latency p99	Latency Max
1127	0.31 req/s	695 ms	2.18 s

Latency p99

STAGE	STATUS	ARCHIVEINFO	RELEASE	CANCEL	DELETE	TAKE_OVER
48.2 ms	840 ms	39.8 ms	4.15 ms	N/A	N/A	1.61 s

Latency max

STAGE	STATUS	ARCHIVEINFO	RELEASE	CANCEL	DELETE	TAKE_OVER
48.2 ms	841 ms	96.4 ms	4.26 ms	N/A	N/A	1.68 s

Latency p99

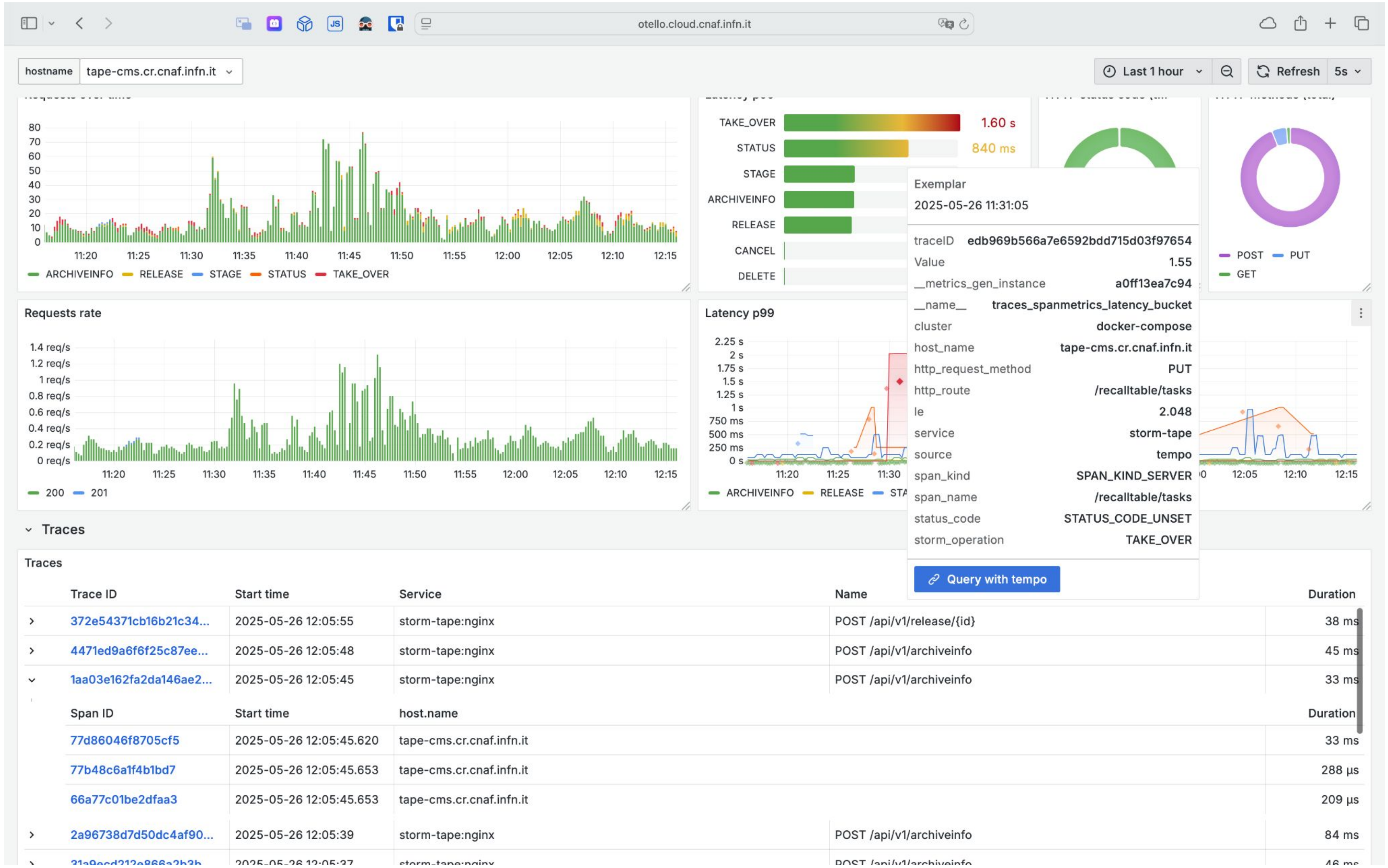
TAKE_OVER	1.61 s
STATUS	840 ms
STAGE	48.2 ms
ARCHIVEINFO	39.8 ms
RELEASE	4.15 ms
CANCEL	N/A
DELETE	N/A

Traces

Traces

Trace ID	Start time	Service	Name	Duration
536cf36d7ab974ba918...	2025-05-26 12:05:02	storm-tape:nginx	POST /api/v1/archiveinfo	53 ms
24e7e97a15c880c96d6...	2025-05-26 12:04:55	storm-tape:nginx	POST /api/v1/archiveinfo	42 ms
5ed589ce935c159d54...	2025-05-26 12:04:40	storm-tape:nginx	GET /api/v1/stage/{id}	1.00 s
563a8ddada872b9104...	2025-05-26 12:04:29	storm-tape:nginx	POST /api/v1/archiveinfo	76 ms

StoRM Tape



hostname: tape-cms.cr.cnaf.infn.it

Exemplar
2025-05-26 11:31:05

traceID	edb969b566a7e6592bdd715d03f97654
Value	1.55
__metrics_gen_instance	a0ff13ea7c94
__name__	traces_spanmetrics_latency_bucket
cluster	docker-compose
host_name	tape-cms.cr.cnaf.infn.it
http_request_method	PUT
http_route	/recalltable/tasks
le	2.048
service	storm-tape
source	tempo
span_kind	SPAN_KIND_SERVER
span_name	/recalltable/tasks
status_code	STATUS_CODE_UNSET
storm_operation	TAKE_OVER

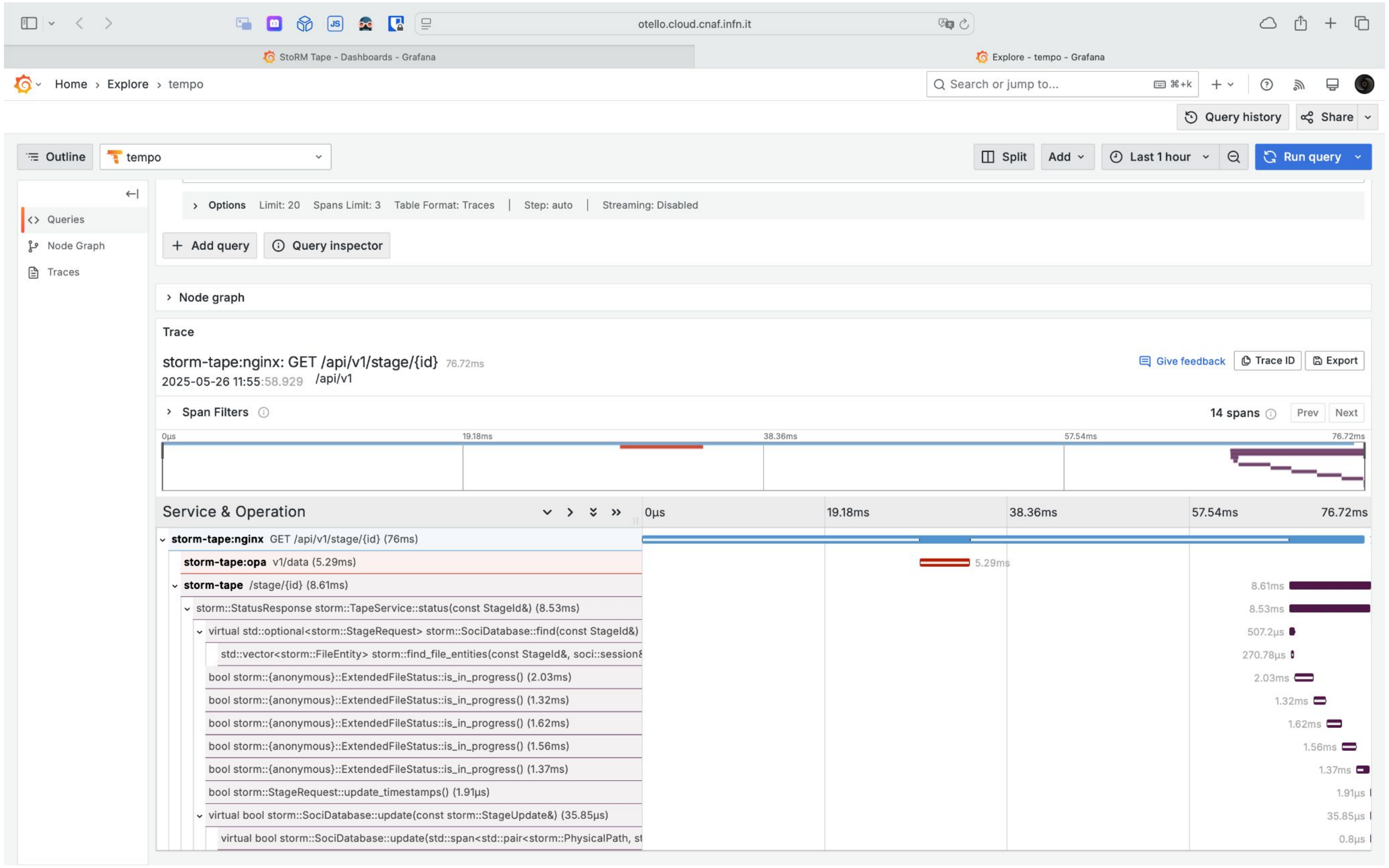
Traces

Trace ID	Start time	Service	Name	Duration
372e54371cb16b21c34...	2025-05-26 12:05:55	storm-tape:nginx	POST /api/v1/release/{id}	38 ms
4471ed9a6f6f25c87ee...	2025-05-26 12:05:48	storm-tape:nginx	POST /api/v1/archiveinfo	45 ms
1aa03e162fa2da146ae2...	2025-05-26 12:05:45	storm-tape:nginx	POST /api/v1/archiveinfo	33 ms

Span ID	Start time	host.name	Duration
77d86046f8705cf5	2025-05-26 12:05:45.620	tape-cms.cr.cnaf.infn.it	33 ms
77b48c6a1f4b1bd7	2025-05-26 12:05:45.653	tape-cms.cr.cnaf.infn.it	288 µs
66a77c01be2dfaa3	2025-05-26 12:05:45.653	tape-cms.cr.cnaf.infn.it	209 µs

Trace ID	Start time	Service	Name	Duration
2a96738d7d50dc4af90...	2025-05-26 12:05:39	storm-tape:nginx	POST /api/v1/archiveinfo	84 ms
31a9ecd212e866a2b3b...	2025-05-26 12:05:37	storm-tape:nginx	POST /api/v1/archiveinfo	46 ms

StoRM Tape



Options Limit: 20 Spans Limit: 3 Table Format: Traces | Step: auto | Streaming: Disabled

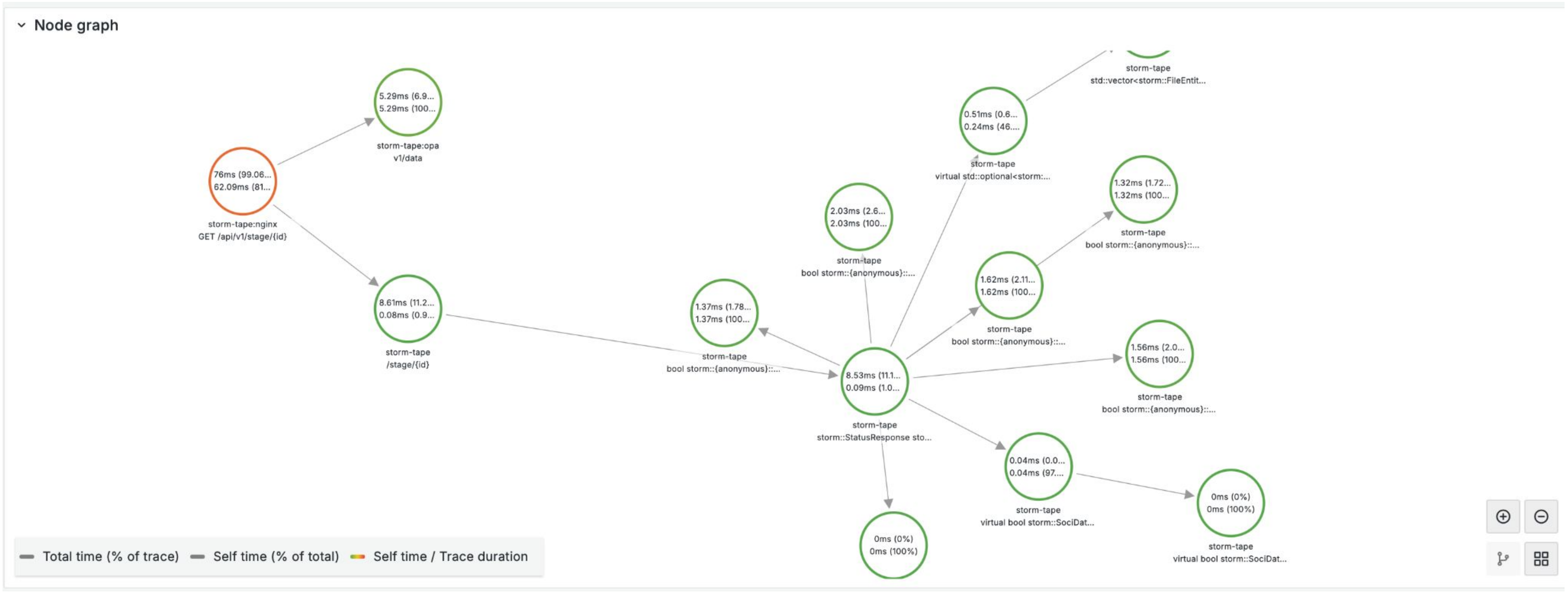
Trace

storm-tape:nginx: GET /api/v1/stage/{id} 76.72ms
2025-05-26 11:55:58.929 /api/v1

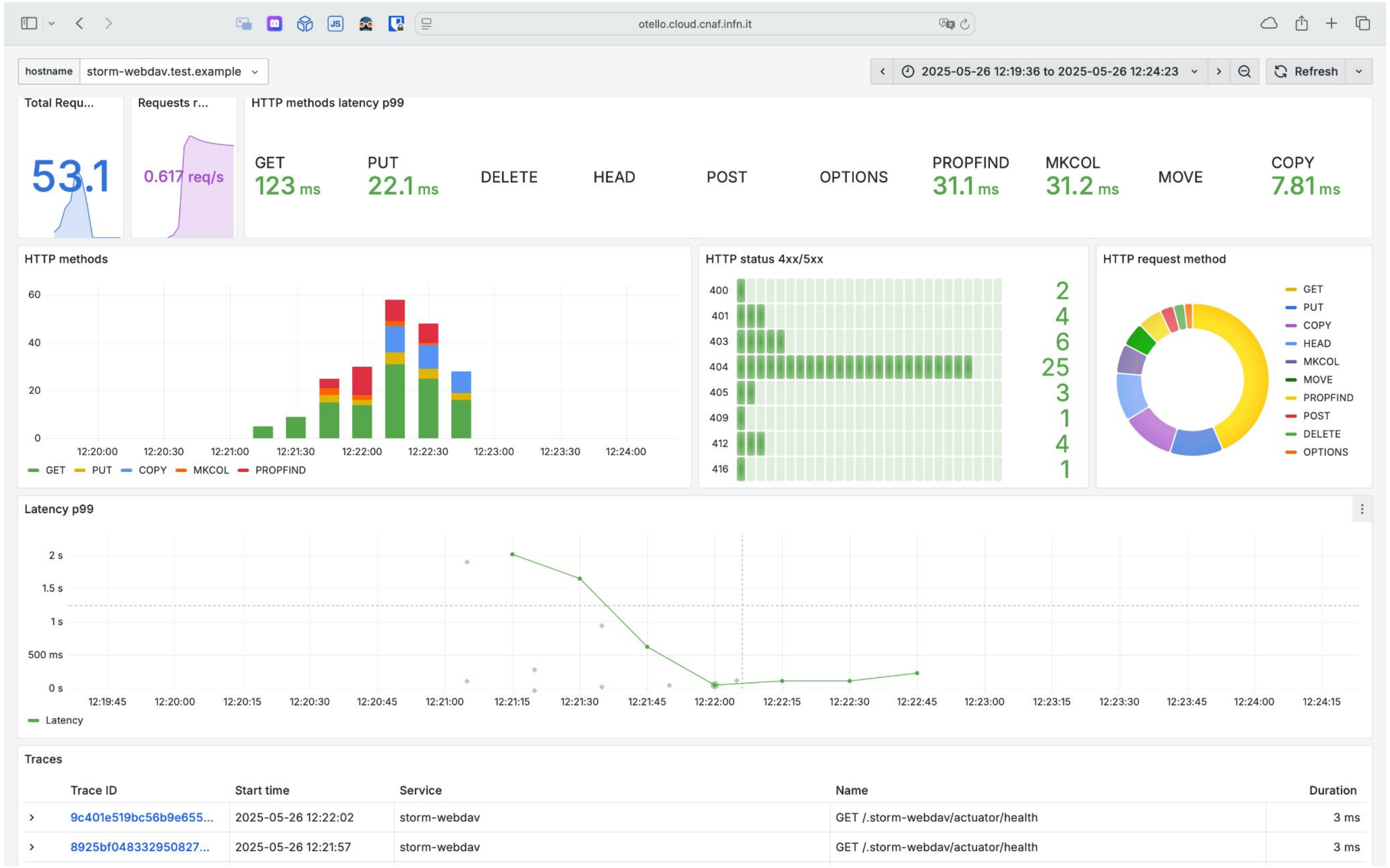
14 spans

```
Service & Operation                                                                         0µs   19.18ms   38.36ms   57.54ms   76.72ms
storm-tape:nginx GET /api/v1/stage/{id} (76ms)
  storm-tape:opa v1/data (5.29ms)                                                                   5.29ms
  storm-tape /stage/{id} (8.61ms)                                                                                         8.61ms
    storm::StatusResponse storm::TapeService::status(const StageId&) (8.53ms)                                             8.53ms
      virtual std::optional<storm::StageRequest> storm::SociDatabase::find(const StageId&)                               507.2µs
        std::vector<storm::FileEntity> storm::find_file_entities(const StageId&, soci::session                           270.78µs
      bool storm::{anonymous}::ExtendedFileStatus::is_in_progress() (2.03ms)                                             2.03ms
      bool storm::{anonymous}::ExtendedFileStatus::is_in_progress() (1.32ms)                                             1.32ms
      bool storm::{anonymous}::ExtendedFileStatus::is_in_progress() (1.62ms)                                             1.62ms
      bool storm::{anonymous}::ExtendedFileStatus::is_in_progress() (1.56ms)                                             1.56ms
      bool storm::{anonymous}::ExtendedFileStatus::is_in_progress() (1.37ms)                                             1.37ms
      bool storm::StageRequest::update_timestamps() (1.91µs)                                                             1.91µs
      virtual bool storm::SociDatabase::update(const storm::StageUpdate&) (35.85µs)                                      35.85µs
        virtual bool storm::SociDatabase::update(std::span<std::pair<storm::PhysicalPath, st                             0.8µs
```

StoRM Tap


Node graph
76ms (99.06...
62.09ms (81...
storm-tape:nginx
GET /api/v1/stage/{id}
5.29ms (6.9...
5.29ms (100...
storm-tape:opa
v1/data
8.61ms (11.2...
0.08ms (0.9...
storm-tape
/stage/{id}
1.37ms (1.78...
1.37ms (100...
storm-tape
bool storm::{anonymous}::...
2.03ms (2.6...
2.03ms (100...
storm-tape
bool storm::{anonymous}::...
0.51ms (0.6...
0.24ms (46....
storm-tape
virtual std::optional<storm:...
storm-tape
std::vector<storm::FileEntit...
1.32ms (1.72...
1.32ms (100...
storm-tape
bool storm::{anonymous}::...
1.62ms (2.11...
1.62ms (100...
storm-tape
bool storm::{anonymous}::...
1.56ms (2.0...
1.56ms (100...
storm-tape
bool storm::{anonymous}::...
8.53ms (11.1...
0.09ms (1.0...
storm-tape
storm::StatusResponse sto...
0.04ms (0.0...
0.04ms (97....
storm-tape
virtual bool storm::SociDat...
0ms (0%)
0ms (100%)
storm-tape
virtual bool storm::SociDat...
0ms (0%)
0ms (100%)
Total time (% of trace)
Self time (% of total)
Self time / Trace duration

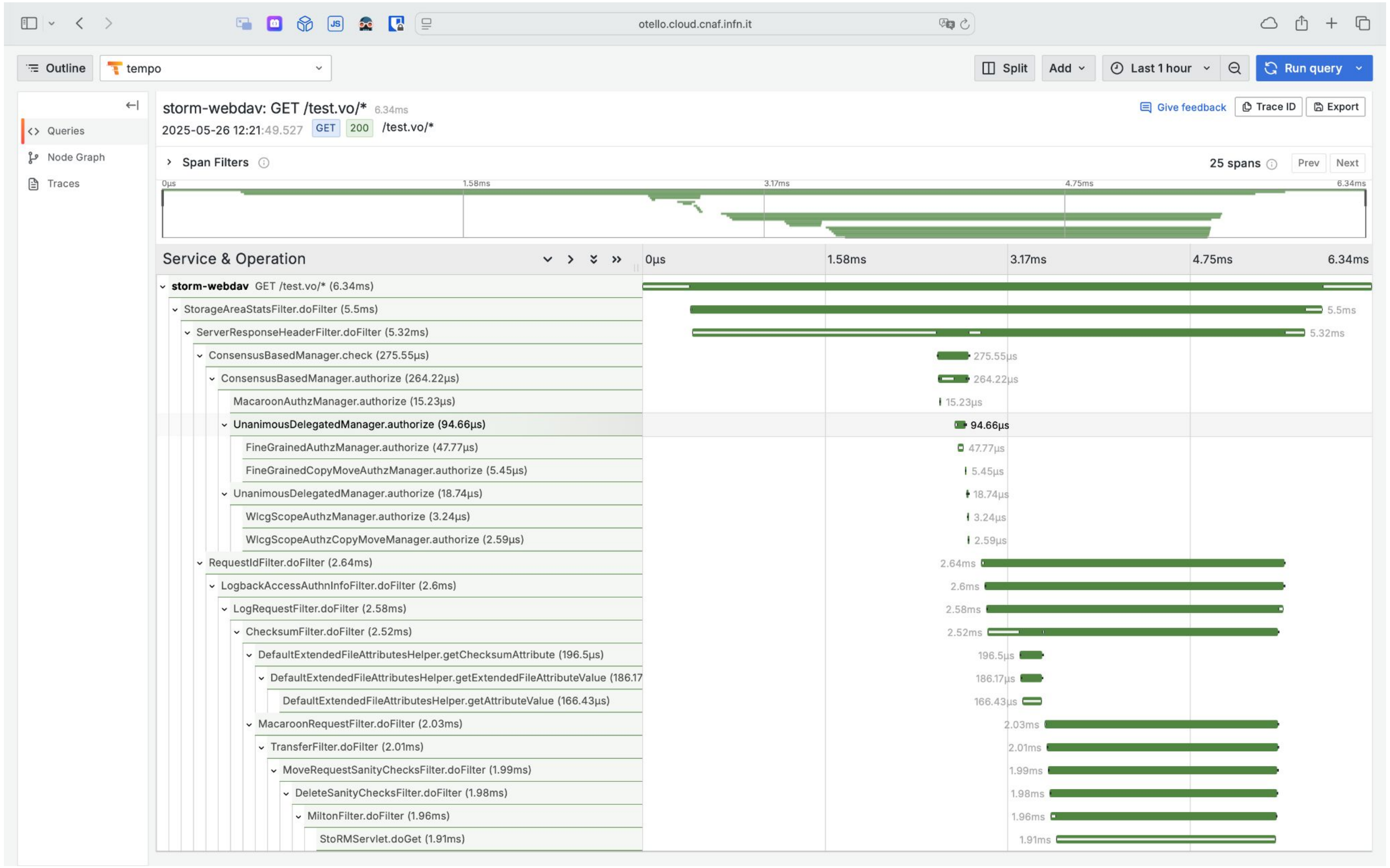
StoRM WebDAV



Traces

Trace ID	Start time	Service	Name	Duration
9c401e519bc56b9e655...	2025-05-26 12:22:02	storm-webdav	GET /.storm-webdav/actuator/health	3 ms
8925bf048332950827...	2025-05-26 12:21:57	storm-webdav	GET /.storm-webdav/actuator/health	3 ms

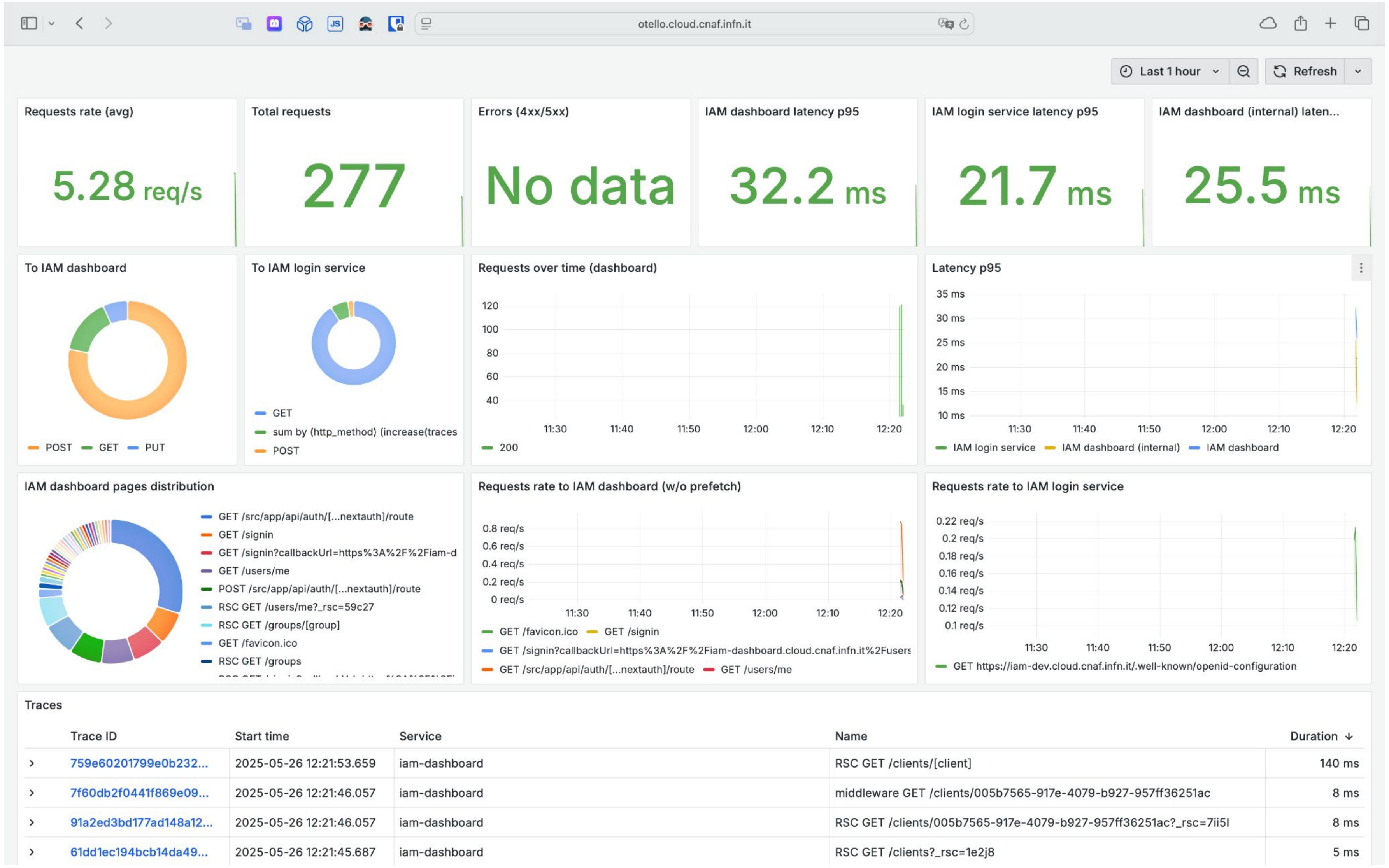
StoRM WebDAV



```
storm-webdav: GET /test.vo/*  6.34ms
2025-05-26 12:21:49.527  GET  200  /test.vo/*

Service & Operation                                                     0µs   1.58ms   3.17ms   4.75ms   6.34ms
storm-webdav GET /test.vo/* (6.34ms)
  StorageAreaStatsFilter.doFilter (5.5ms)                                                         5.5ms
    ServerResponseHeaderFilter.doFilter (5.32ms)                                                  5.32ms
      ConsensusBasedManager.check (275.55µs)                                                      275.55µs
        ConsensusBasedManager.authorize (264.22µs)                                                264.22µs
          MacaroonAuthzManager.authorize (15.23µs)                                                15.23µs
          UnanimousDelegatedManager.authorize (94.66µs)                                           94.66µs
            FineGrainedAuthzManager.authorize (47.77µs)                                           47.77µs
            FineGrainedCopyMoveAuthzManager.authorize (5.45µs)                                    5.45µs
          UnanimousDelegatedManager.authorize (18.74µs)                                           18.74µs
            WlcgScopeAuthzManager.authorize (3.24µs)                                              3.24µs
            WlcgScopeAuthzCopyMoveManager.authorize (2.59µs)                                      2.59µs
      RequestIdFilter.doFilter (2.64ms)                                                           2.64ms
        LogbackAccessAuthnInfoFilter.doFilter (2.6ms)                                             2.6ms
          LogRequestFilter.doFilter (2.58ms)                                                      2.58ms
            ChecksumFilter.doFilter (2.52ms)                                                      2.52ms
              DefaultExtendedFileAttributesHelper.getChecksumAttribute (196.5µs)                  196.5µs
                DefaultExtendedFileAttributesHelper.getExtendedFileAttributeValue (186.17         186.17µs
                  DefaultExtendedFileAttributesHelper.getAttributeValue (166.43µs)                166.43µs
              MacaroonRequestFilter.doFilter (2.03ms)                                             2.03ms
                TransferFilter.doFilter (2.01ms)                                                  2.01ms
                  MoveRequestSanityChecksFilter.doFilter (1.99ms)                                 1.99ms
                    DeleteSanityChecksFilter.doFilter (1.98ms)                                    1.98ms
                      MiltonFilter.doFilter (1.96ms)                                              1.96ms
                        StoRMServlet.doGet (1.91ms)                                               1.91ms
```

IAM Dashboard

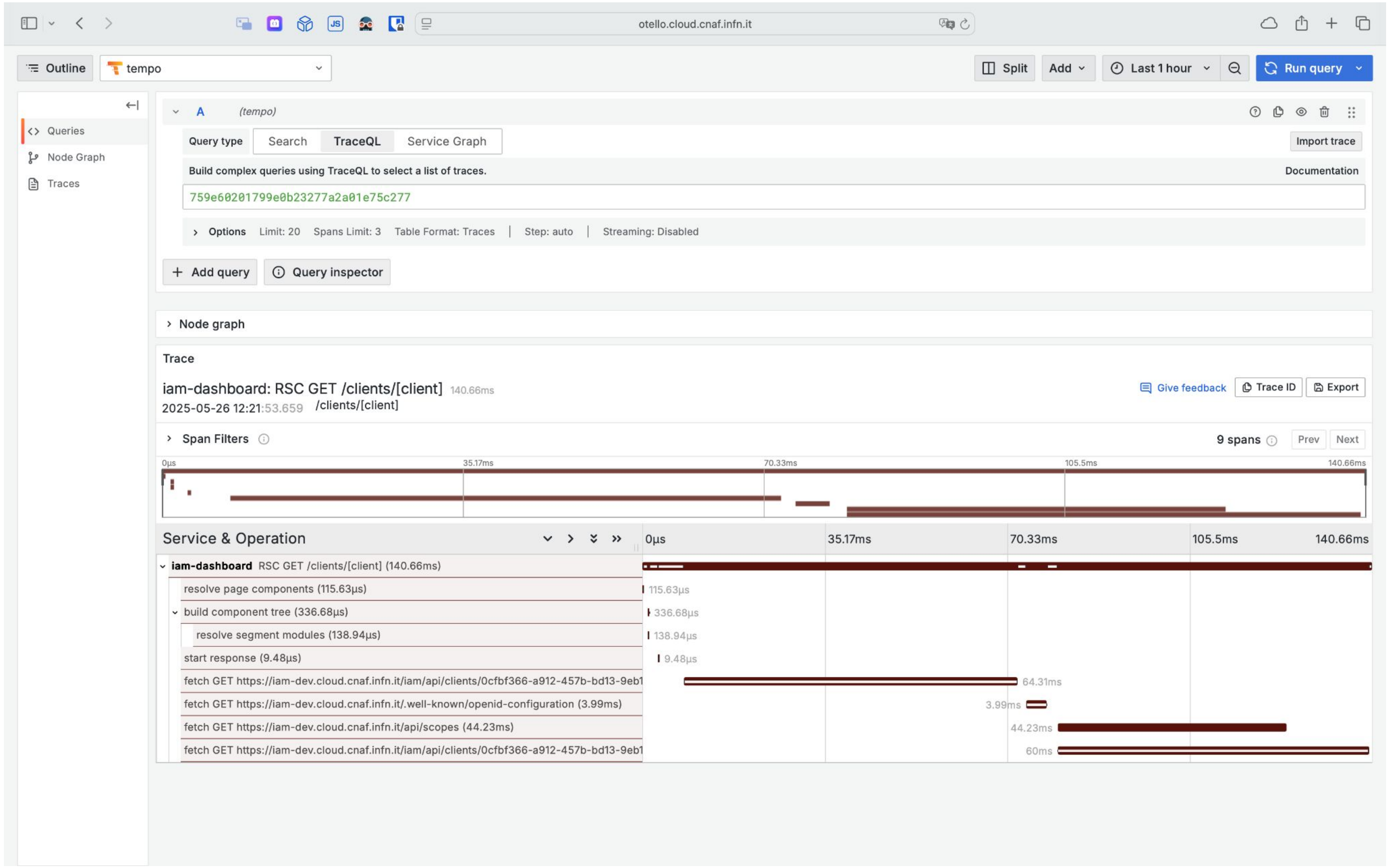

otello.cloud.cnaf.infn.it
Last 1 hour
Refresh
To IAM dashboard
POST
GET
PUT
To IAM login service
GET
sum by (http_method) (increase(traces
POST
Requests over time (dashboard)
120
100
80
60
40
11:30
11:40
11:50
12:00
12:10
12:20
200
Latency p95
35 ms
30 ms
25 ms
20 ms
15 ms
10 ms
11:30
11:40
11:50
12:00
12:10
12:20
IAM login service
IAM dashboard (internal)
IAM dashboard
IAM dashboard pages distribution
GET /src/app/api/auth/[...nextauth]/route
GET /signin
GET /signin?callbackUrl=https%3A%2F%2Fiam-d
GET /users/me
POST /src/app/api/auth/[...nextauth]/route
RSC GET /users/me?_rsc=59c27
RSC GET /groups/[group]
GET /favicon.ico
RSC GET /groups
Requests rate to IAM dashboard (w/o prefetch)
0.8 req/s
0.6 req/s
0.4 req/s
0.2 req/s
0 req/s
11:30
11:40
11:50
12:00
12:10
12:20
GET /favicon.ico
GET /signin
GET /signin?callbackUrl=https%3A%2F%2Fiam-dashboard.cloud.cnaf.infn.it%2Fusers
GET /src/app/api/auth/[...nextauth]/route
GET /users/me
Requests rate to IAM login service
0.22 req/s
0.2 req/s
0.18 req/s
0.16 req/s
0.14 req/s
0.12 req/s
0.1 req/s
11:30
11:40
11:50
12:00
12:10
12:20
GET https://iam-dev.cloud.cnaf.infn.it/.well-known/openid-configuration

Requests rate (avg)	Total requests	Errors (4xx/5xx)	IAM dashboard latency p95	IAM login service latency p95	IAM dashboard (internal) laten...
5.28 req/s	277	No data	32.2 ms	21.7 ms	25.5 ms

Traces

Trace ID	Start time	Service	Name	Duration ↓
759e60201799e0b232...	2025-05-26 12:21:53.659	iam-dashboard	RSC GET /clients/[client]	140 ms
7f60db2f0441f869e09...	2025-05-26 12:21:46.057	iam-dashboard	middleware GET /clients/005b7565-917e-4079-b927-957ff36251ac	8 ms
91a2ed3bd177ad148a12...	2025-05-26 12:21:46.057	iam-dashboard	RSC GET /clients/005b7565-917e-4079-b927-957ff36251ac?_rsc=7ii5l	8 ms
61dd1ec194bcb14da49...	2025-05-26 12:21:45.687	iam-dashboard	RSC GET /clients?_rsc=1e2j8	5 ms

IAM Dashboard



Query type: TraceQL

Build complex queries using TraceQL to select a list of traces.

```
759e60201799e0b23277a2a01e75c277
```

Options Limit: 20 Spans Limit: 3 Table Format: Traces | Step: auto | Streaming: Disabled

Trace

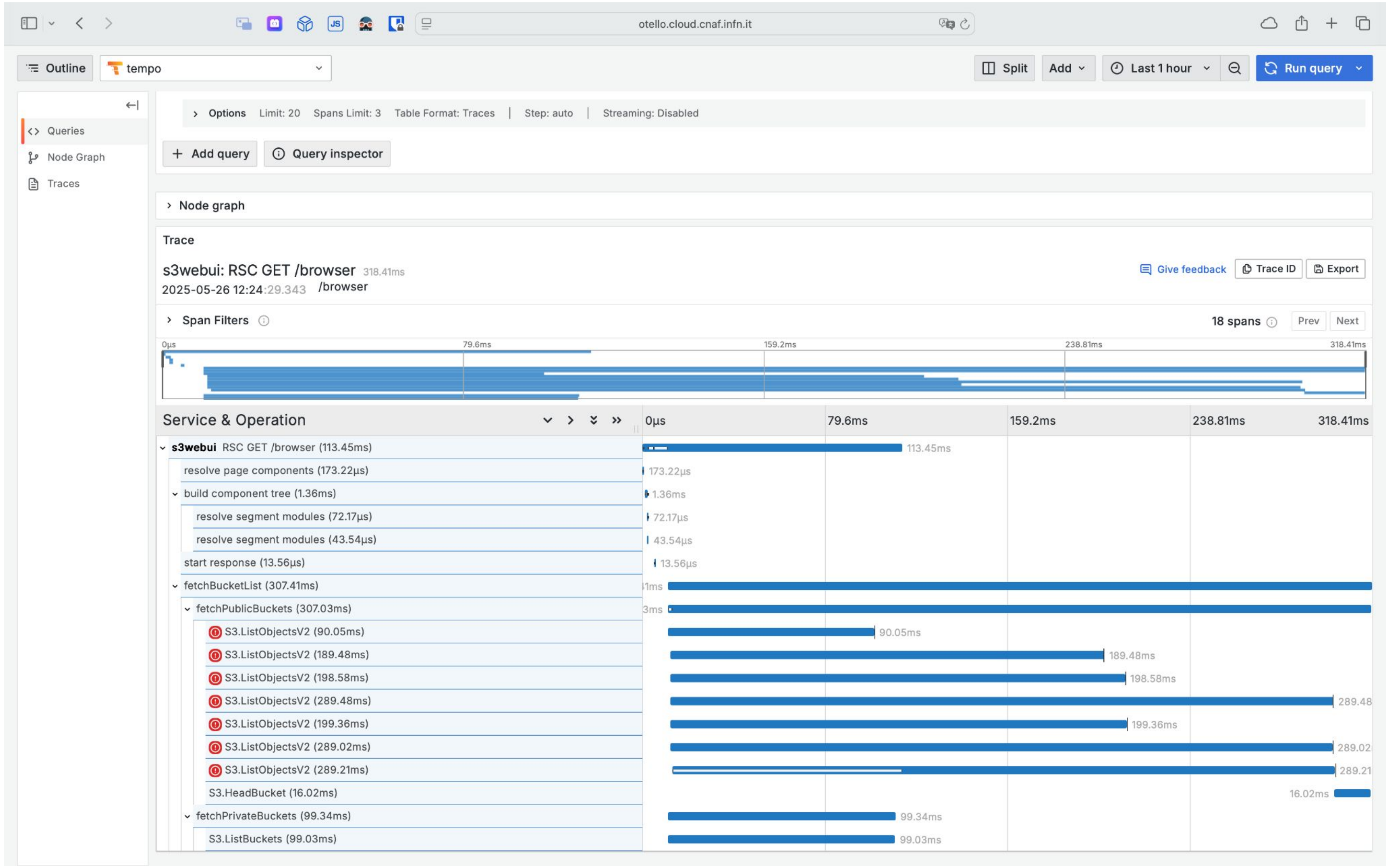
iam-dashboard: RSC GET /clients/[client] 140.66ms
2025-05-26 12:21:53.659 /clients/[client]

Span Filters — 9 spans

0µs 35.17ms 70.33ms 105.5ms 140.66ms

```
Service & Operation                                                                    0µs   35.17ms   70.33ms   105.5ms   140.66ms
iam-dashboard RSC GET /clients/[client] (140.66ms)
  resolve page components (115.63µs)                                                   115.63µs
  build component tree (336.68µs)                                                      336.68µs
    resolve segment modules (138.94µs)                                                 138.94µs
  start response (9.48µs)                                                              9.48µs
  fetch GET https://iam-dev.cloud.cnaf.infn.it/iam/api/clients/0cfbf366-a912-457b-bd13-9eb1   64.31ms
  fetch GET https://iam-dev.cloud.cnaf.infn.it/.well-known/openid-configuration (3.99ms)    3.99ms
  fetch GET https://iam-dev.cloud.cnaf.infn.it/api/scopes (44.23ms)                    44.23ms
  fetch GET https://iam-dev.cloud.cnaf.infn.it/iam/api/clients/0cfbf366-a912-457b-bd13-9eb1   60ms
```

S3 WebUI


otello.cloud.cnaf.infn.it

Options Limit: 20 Spans Limit: 3 Table Format: Traces | Step: auto | Streaming: Disabled

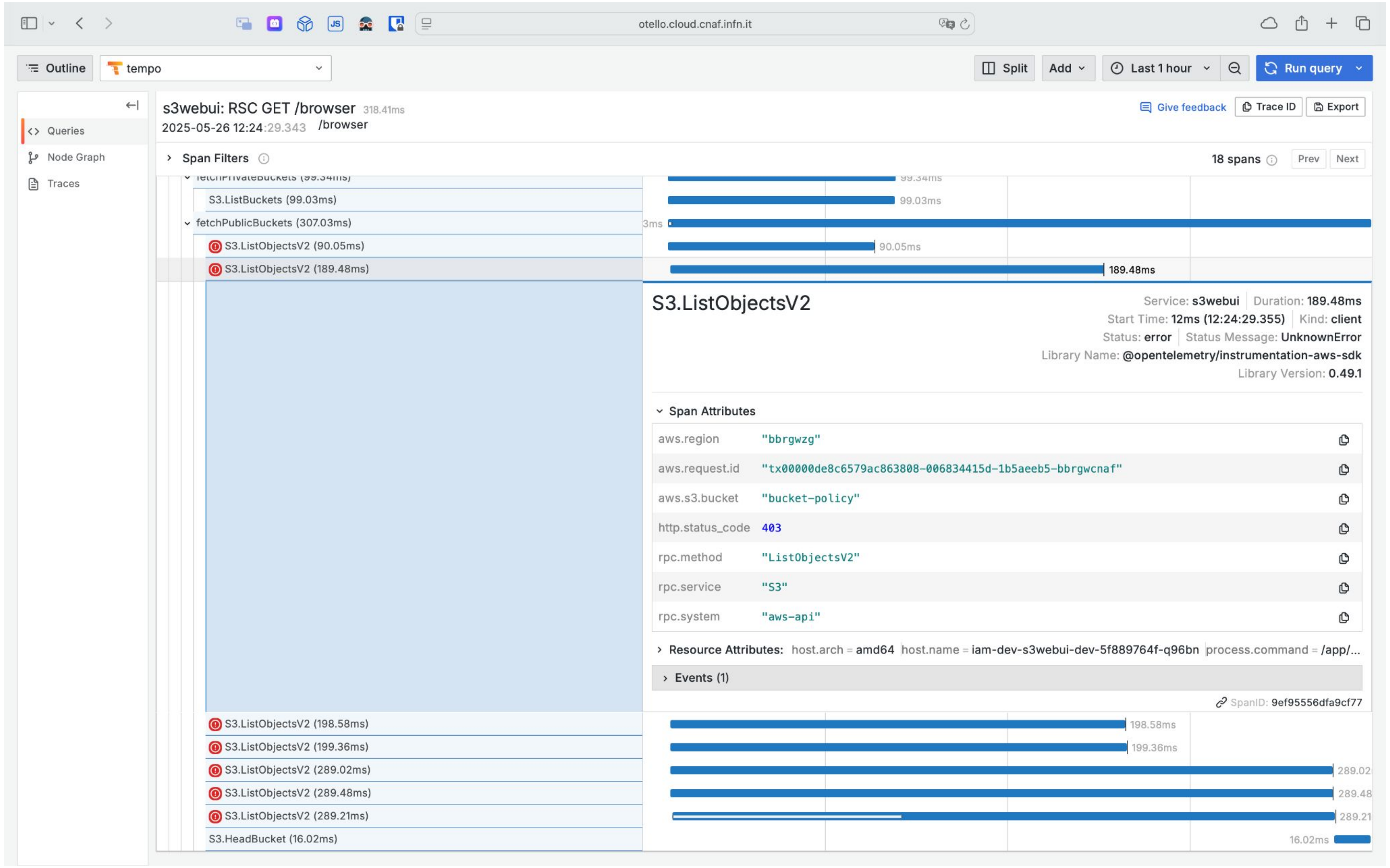
Trace

s3webui: RSC GET /browser 318.41ms
2025-05-26 12:24:29.343 /browser

18 spans

Service & Operation	Duration
s3webui RSC GET /browser (113.45ms)	113.45ms
resolve page components (173.22µs)	173.22µs
build component tree (1.36ms)	1.36ms
resolve segment modules (72.17µs)	72.17µs
resolve segment modules (43.54µs)	43.54µs
start response (13.56µs)	13.56µs
fetchBucketList (307.41ms)	
fetchPublicBuckets (307.03ms)	
S3.ListObjectsV2 (90.05ms)	90.05ms
S3.ListObjectsV2 (189.48ms)	189.48ms
S3.ListObjectsV2 (198.58ms)	198.58ms
S3.ListObjectsV2 (289.48ms)	289.48
S3.ListObjectsV2 (199.36ms)	199.36ms
S3.ListObjectsV2 (289.02ms)	289.02
S3.ListObjectsV2 (289.21ms)	289.21
S3.HeadBucket (16.02ms)	16.02ms
fetchPrivateBuckets (99.34ms)	99.34ms
S3.ListBuckets (99.03ms)	99.03ms

S3 WebUI



otello.cloud.cnaf.infn.it

s3webui: RSC GET /browser 318.41ms
2025-05-26 12:24:29.343 /browser

Span Filters — 18 spans

```
  fetchPrivateBuckets (99.34ms)              99.34ms
    S3.ListBuckets (99.03ms)                 99.03ms
  fetchPublicBuckets (307.03ms)
    S3.ListObjectsV2 (90.05ms)               90.05ms
    S3.ListObjectsV2 (189.48ms)              189.48ms
    S3.ListObjectsV2 (198.58ms)              198.58ms
    S3.ListObjectsV2 (199.36ms)              199.36ms
    S3.ListObjectsV2 (289.02ms)              289.02
    S3.ListObjectsV2 (289.48ms)              289.48
    S3.ListObjectsV2 (289.21ms)              289.21
    S3.HeadBucket (16.02ms)                  16.02ms
```

S3.ListObjectsV2

Service: s3webui | Duration: 189.48ms
Start Time: 12ms (12:24:29.355) | Kind: client
Status: error | Status Message: UnknownError
Library Name: @opentelemetry/instrumentation-aws-sdk
Library Version: 0.49.1

Span Attributes

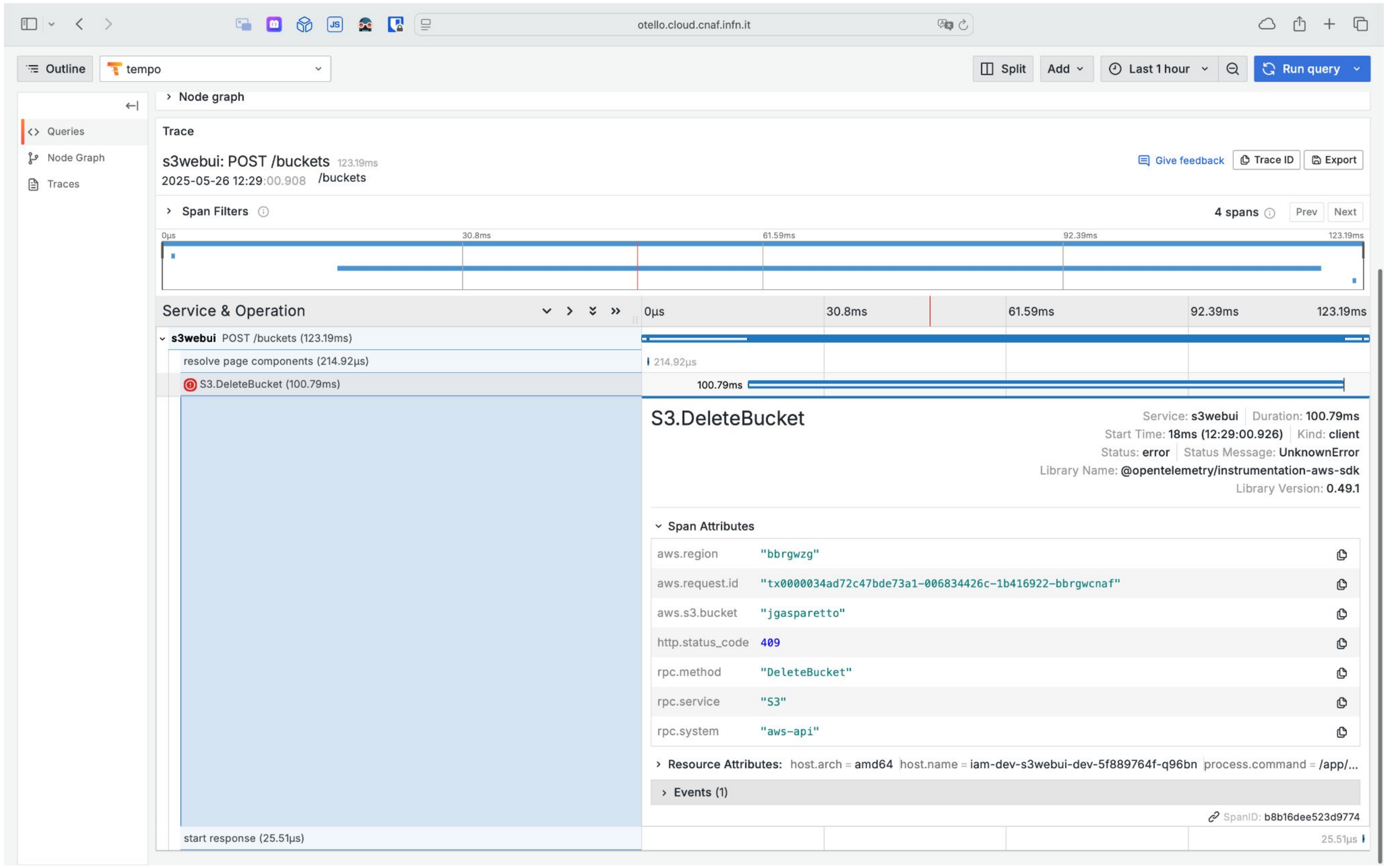
aws.region	"bbrgwzg"
aws.request.id	"tx00000de8c6579ac863808-006834415d-1b5aeeb5-bbrgwcnaf"
aws.s3.bucket	"bucket-policy"
http.status_code	403
rpc.method	"ListObjectsV2"
rpc.service	"S3"
rpc.system	"aws-api"

Resource Attributes: host.arch = amd64 host.name = iam-dev-s3webui-dev-5f889764f-q96bn process.command = /app/...

Events (1)

SpanID: 9ef95556dfa9cf77

S3 WebUI



s3webui: POST /buckets 123.19ms
2025-05-26 12:29:00.908 /buckets

s3webui POST /buckets (123.19ms)
resolve page components (214.92µs)
S3.DeleteBucket (100.79ms)
start response (25.51µs)

S3.DeleteBucket

Service: s3webui | Duration: 100.79ms
Start Time: 18ms (12:29:00.926) | Kind: client
Status: error | Status Message: UnknownError
Library Name: @opentelemetry/instrumentation-aws-sdk
Library Version: 0.49.1

Span Attributes

aws.region	"bbrgwzg"
aws.request.id	"tx0000034ad72c47bde73a1-006834426c-1b416922-bbrgwcnaf"
aws.s3.bucket	"jgasparetto"
http.status_code	409
rpc.method	"DeleteBucket"
rpc.service	"S3"
rpc.system	"aws-api"

Resource Attributes: host.arch = amd64 host.name = iam-dev-s3webui-dev-5f889764f-q96bn process.command = /app/...

Events (1)

SpanID: b8b16dee523d9774

Conclusions

- OpenTelemetry is a de-facto standard to produce telemetry data across a distributed system of any size.
- Tests (Otello) demonstrated that this technology is really promising for monitoring and debugging, especially to identify bottlenecks and unwanted path in our code.
- Even though Otello is experimental, StoRM Tape is already in production with OpenTelemetry support.
- The goal is to extend OpenTelemetry support to all software developed at INFN-CNAF, merging the Otello stack with the t1metria monitoring service.
- Proposal: integration of the OpenTelemetry support to software produced for DataCloud, perhaps integrating a stack similar to Otello and managed by WP1

Backup

Propagation

- A request can travel across several distinct services, such as, for example, a reverse proxy (NGINX), an authorization engine (OPA), a REST API, a database (SQL), etc.
- OpenTelemetry fully supports the concept of Context Propagation as indicated by the W3C TraceContext specification, which *“allows traces to build causal information about a system across services that are arbitrarily distributed across process and network boundaries”*.
- This mechanism is generally established in web services using the **HTTP headers** `traceparent` and `tracestate`
- Spans are always emitted individually to the collector by the different services.
- Many third-party software, such as OPA, NGINX and K8S Ingress Controller, natively supports OpenTelemetry’s Context Propagation

Sampling

- The amount of telemetry data, especially traces, can be huge, and overhead at runtime can be non-negligible. *"Sampling is one of the most effective ways to reduce the costs of observability without losing visibility"* https://opentelemetry.io/docs/concepts/sampling.
- Sampling can be performed probabilistically, for example: "sample only 10% of the incoming requests"
- Sampling can be controlled via `ParentBasedSampler`s that produce a span only if a parent span is detected.
- The two approaches can be combined. For example, the first service receiving the requests (e.g, NGINX) can decide to sample only a fraction of them, then the other services of the chain will sample or based on the presence or not of the parent span