OpenTelemetry 101

Zameer Fouzan
Senior Developer Relations Engineer

© 2023 New Relic, Inc. All rights reserved

Agenda

01	OpenTelemetry Background
02	Core Concepts of OpenTelemetry
03	Instrumentation - Hands-on
04	Open Telemetry Collector

© 2023 New Relic, Inc. All rights reserved.

WORKED FINE
IN DEV
OPS PROBLEM NOW

OpenTelemetry

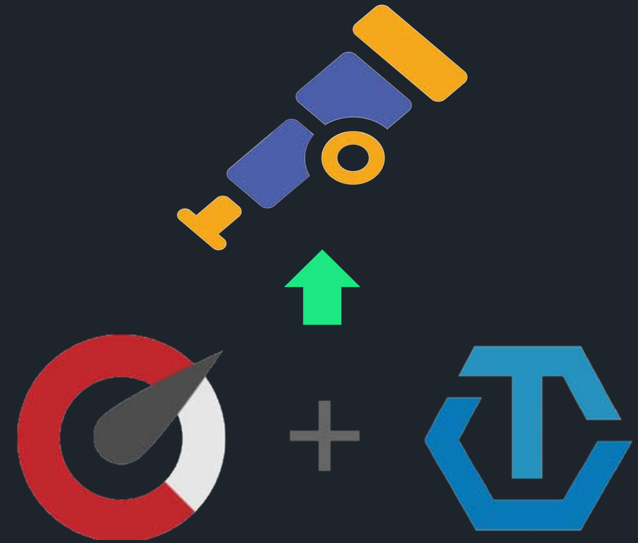
Background

© 2023 New Relic, Inc. All rights reserved.

OpenTelemetry

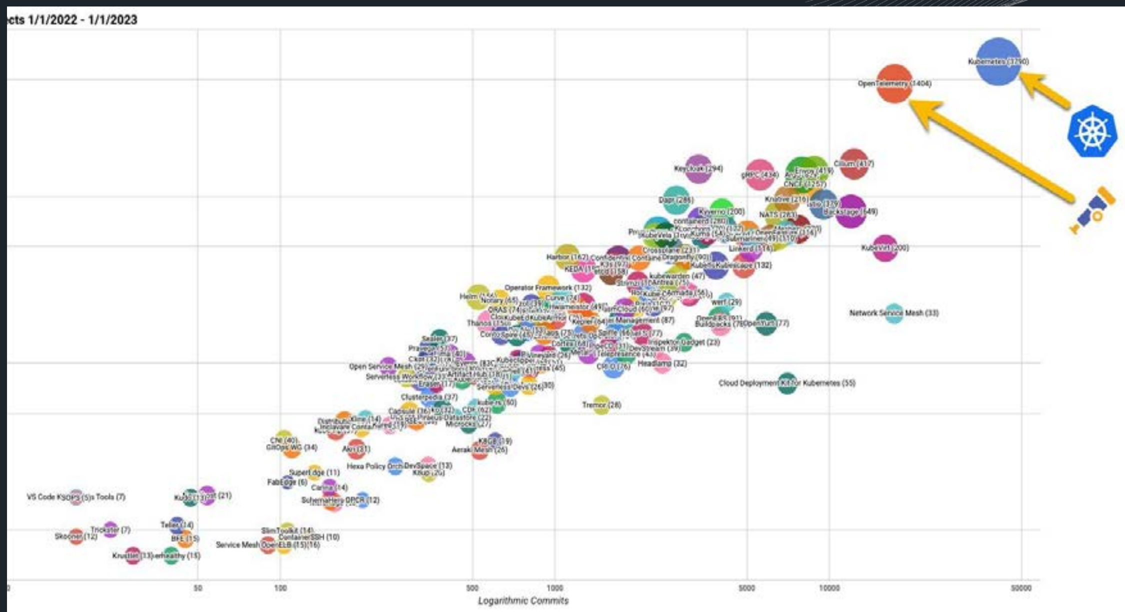
Key Facts on OpenTelemetry

- OpenTelemetry is an **Incubating** project of CNCF.
- Formed through a merger of the **OpenTracing** and **OpenCensus** projects.
- **Vendor agnostic** - set of APIs, libraries, integrations, and a collector service for telemetry.
- **Standardizes** how you collect telemetry data from your applications and services.
- Send it to an **Observability platform** of your choice.



© 2023 New Relic, Inc. All rights reserved.

The Rise of OpenTelemetry

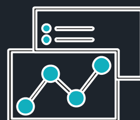


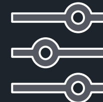
OpenTelemetry is **second** in CNCF - strong interest in modern **Observability**.

Enhanced support for **Open Standards**, including OpenTelemetry, eBPF, and Grafana - Gartner MQ 2022

© 2023 New Relic, Inc. All rights reserved.

The Rise of OpenTelemetry





Ubiquity

Promotes better coverage for instrumentation

Vendor Neutral

Provides flexibility to change backend

Interoperable

End-to-end visibility with standard instrumentation

Configurable

Pick and choose from the pieces what is needed

By 2025, **70%** of new cloud-native application monitoring will use open source instrumentation

(1) Source: Gartner Magic Quadrant 2021 for Application Performance Monitoring - link

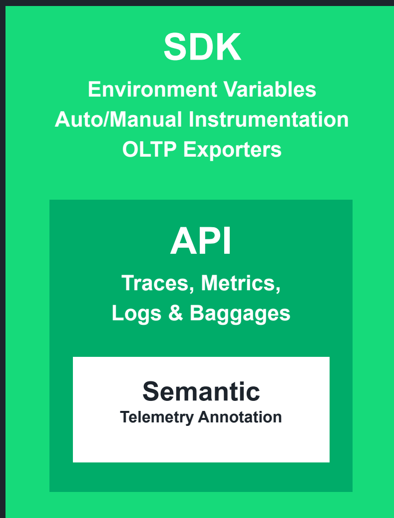
© 2023 New Relic, Inc. All rights reserved.

Core Concepts

The building blocks

© 2023 New Relic, Inc. All rights reserved.

OpenTelemetry - Building blocks



Core Concepts on Instrumentation

- **Semantic Conventions** - annotate telemetry with attributes specific to the represented operation, such as HTTP calls.
- **API** - data types for tracing, metrics, and logging data.
- **SDK** - language-specific implementation of the API.
- SDKs incorporate automatic instrumentation for common libraries and frameworks for your application.
- **OpenTelemetry Protocol (OTLP)** - used to send data to your backend Observability platform of choice.

© 2022 New Relic, Inc. All rights reserved.

Application Instrumentation

Hands-On+Demo

© 2023 New Relic, Inc. All rights reserved.

new relic

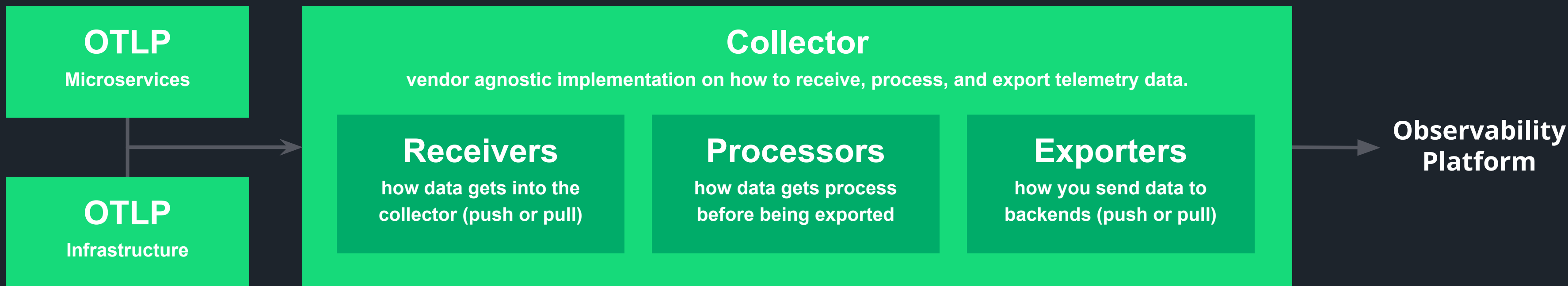
Infrastructure

The Collector

© 2023 New Relic, Inc. All rights reserved.

new relic

OpenTelemetry - Collector



a **proxy** that process multiple telemetry formats, via an **agent** or a **gateway**
(e.g., OTLP, Jaeger, Prometheus, as well as many commercial/proprietary tools)

© 2022 New Relic, Inc. All rights reserved.

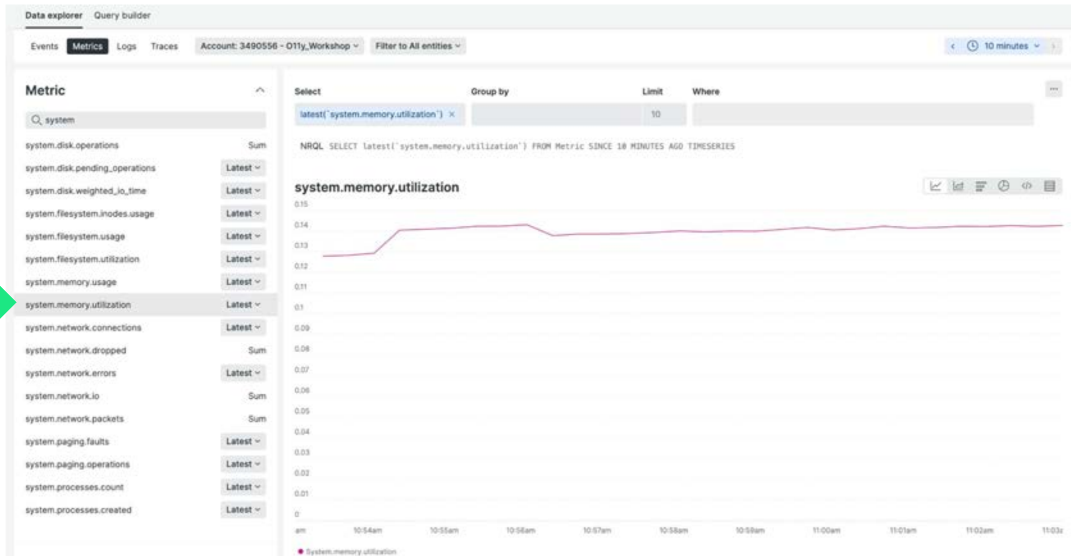
new relic

OpenTelemetry - Collector

```
▸ images
▸ istio-manifests
▸ kubernetes-manifests
▸ otel-kubernetes-manife...
▸ pb
▸ release
▾ src
  ▸ adservice
  ▸ cartservice
  ▸ checkoutservice
  ▸ currencyservice
  ▸ emailservice
  ▸ frontend
  ▸ loadgenerator
  ▾ otel-collector
      Dockerfile
      collector.yaml
  ▾ otel-collector-agent
      Dockerfile
      collector.yaml
  ▸ otel-collector-agent-re...
  ▸ paymentservice
  ▸ productcatalogservice
  ▸ quoteservice
  ▸ recommendationservice
  ▸ shippingservice
  .gitignore
CODEOWNERS
CODE_OF_CONDUCT.md
CONTRIBUTING.md
LICENSE
README.md
SECURITY.md
```

```yaml
receivers:
  hostmetrics:
    collection_interval: 20s
    scrapers:
      cpu:
        metrics:
          system.cpu.utilization:
            enabled: true
      load:
      memory:
        metrics:
          system.memory.utilization:
            enabled: true
      disk:
      filesystem:
        metrics:
          system.filesystem.utilization:
            enabled: true
      network:
      paging:
        metrics:
          system.paging.utilization:
            enabled: true
      processes:
  otlp:
    protocols:
      grpc:

processors:
  batch:
  cumulativetodelta:
    metrics:
      - system.network.io
      - system.disk.operations
      - system.network.dropped
      - system.network.packets
      - process.cpu.time
```



Collect additional infrastructure metrics for OpenTelemtry

OpenTelemetry - Collector

Storing all tracing data is **costly**. Most of the time, you **ONLY NEED THE RIGHT DATA**, when outages are occurring, or patterns are emerging.

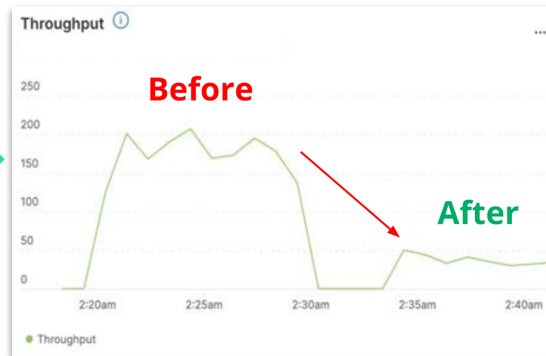
Head based sampling works well for an overall statistical sampling of requests through a distributed system.

Tail based sampling is best to decide what to keep, based on isolated, independent portions of the trace data.

OpenTelemetry - Collector

- src
  - adservice
  - cartservice
  - checkoutservice
  - currencyservice
  - emailservice
  - frontend
  - loadgenerator
  - otel-collector
    - Dockerfile
    - collector.yaml
  - otel-collector-agent
    - Dockerfile
    - collector.yaml
  - otel-collector-agent-re...

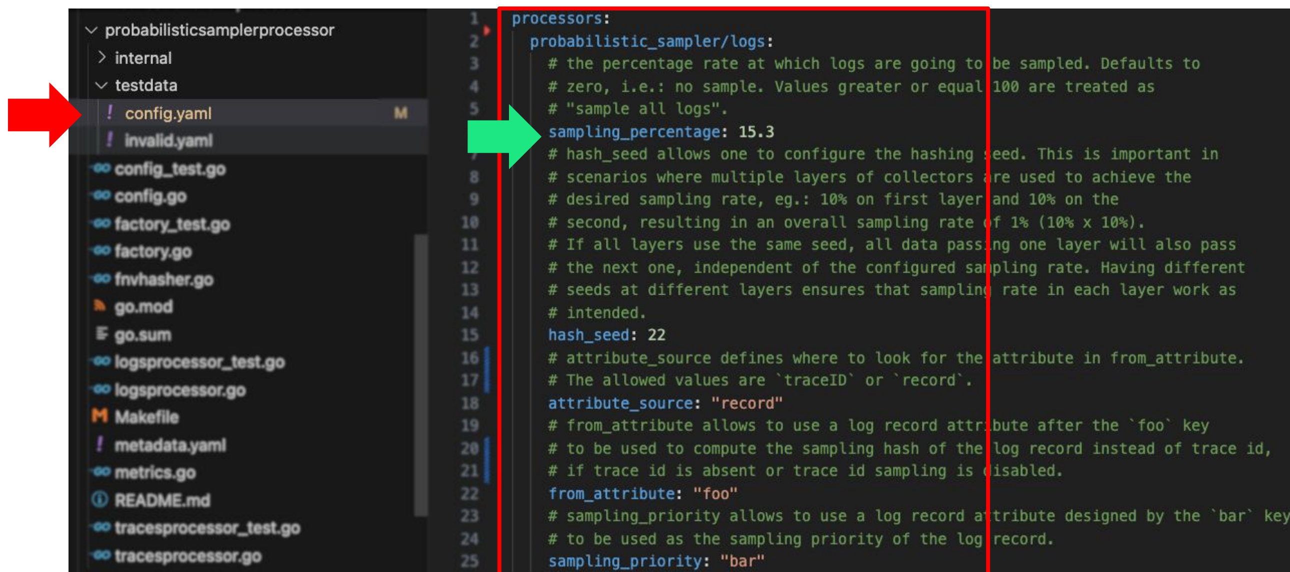
```
30  processors:
31    tail_sampling:
32      decision_wait: 15s
33      num_traces: 100
34      expected_new_traces_per_sec: 10
35      policies:
36        [
37          {
38            name: test-policy-1,
39            type: always_sample
40          },
41          {
42            name: test-policy-2,
43            type: latency,
44            latency: {threshold_ms: 1000}
45          },
46        ]
```



Effects from Tail Based Sampling

Apply processing decision in the Collector, before shipping data out

OpenTelemetry - Collector



Probabilistic Based Sampling

Severless

AWS Lambda

© 2023 New Relic, Inc. All rights reserved.

Lambda Instrumentation

```
"@opentelemetry/instrumentation-aws-lambda": {
  enabled: true,
  disableAwsContextPropagation: true,
  requestHook: (span, { event, context }) => {
    span.setAttribute("faas.name", context.functionName);

    if (event.requestContext && event.requestContext.http) {
      span.setAttribute(
        "faas.http.method",
        event.requestContext.http.method
      );
      span.setAttribute(
        "faas.http.target",
        event.requestContext.http.path
      );
    }
  },
  responseHook: (span, { err, res }) => {
    if (err instanceof Error)
      span.setAttribute("faas.error", err.message);
    if (res) {
      span.setAttribute("faas.http.status_code", res.statusCode);
    }
  },
},
```

- Official Lambda library
- Request Hooks
- Response Hooks
- Custom Attributes

```
NODE_OPTIONS: "--require ./otel-wrapper"
```

© 2023 New Relic, Inc. All rights reserved.

Lambda Instrumentation

otel-sls-sdk-dev-api

Node-Lambda-Otel-SDK | November 19 at 5:31pm | Trace ID: 0b93397ddcaccf11a8ab04ff24deeea1 | Spans: 22

Trace details Logs (0)

Expand all Collapse all Standard Manual Reset Minimize

otel-sls-sdk-dev-api
Node-Lambda-Otel-SDK 59.74 ms

11 GET /api/weather
node-express-otel 20.15 ms

middleware - urlencodedParser 0.02 ms
middleware - <anonymous> 0.03 ms
router - /api/weather 0.01 ms
request handler - / 0.00 ms
middleware - query
middleware - expressInit 0.05 ms
middleware - jsonParser 0.03 ms
GET 18.28 ms
Uninstrumented time 8.58 ms

GET /api/weather
node-express-otel

Performance Attributes Details 20.15 ms DURATION

http.host
3.230.230.121

http.method
GET

http.route
/api/weather

http.scheme
http

http.status_code
200

http.status_text
OK

http.target
/api/weather?location=bangalore

http.url
http://3.230.230.121/api/weather?location=bangalore

http.user_agent
axios/1.6.1

id
304ba16ee5677b62

instrumentation.provider
opentelemetry

Start querying your data

© 2023 New Relic, Inc. All rights reserved.

Recap and Highlights

Exciting times for Open Source Observability!

OpenTelemetry is growing and being adopted at a rapid pace.

Be mindful with your maturity, and plan ahead on the adoption of OpenTelemetry.

Just having telemetry is **NOT** observability. Instrumentation should include contextual **traces**, **logs**, or **metrics** to improve observations.

Deployment of the OpenTelemetry collector is easier to gather Telemetry.

Gather telemetry from your pipelines to measure your MTTI, MTTD, MTTR

Actively investing in OpenTelemetry, and help engineers work based on **"data, not opinions"**.

© 2023 New Relic, Inc. All rights reserved.

Resources

Further Reading

Open Telemetry Resources

- Application Instrumentation
- Open Telemetry on Serverless
- Open Telemetry in CI/CD
- Collector
  - Sampling Strategies
- Collector on Kubernetes
  - Configuration - https://opentelemetry.io/docs/collector/configuration/
  - Helm - https://opentelemetry.io/docs/kubernetes/helm/collector/
  - Operator - https://opentelemetry.io/docs/kubernetes/operator/
    - Automatic instrumentation - https://opentelemetry.io/docs/kubernetes/operator/automatic/

© 2023 New Relic, Inc. All rights reserved.

$whoami

Zameer Fouzan
Senior Developer Relations Engineer

© 2023 New Relic, Inc. All rights reserved.