OpenTelemetry 101

Zameer Fouzan

Senior Developer Relations Engineer

Agenda

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


WORKED FINE
IN DEV
OPS PROBLEM NOW

OpenTelemetry

Background

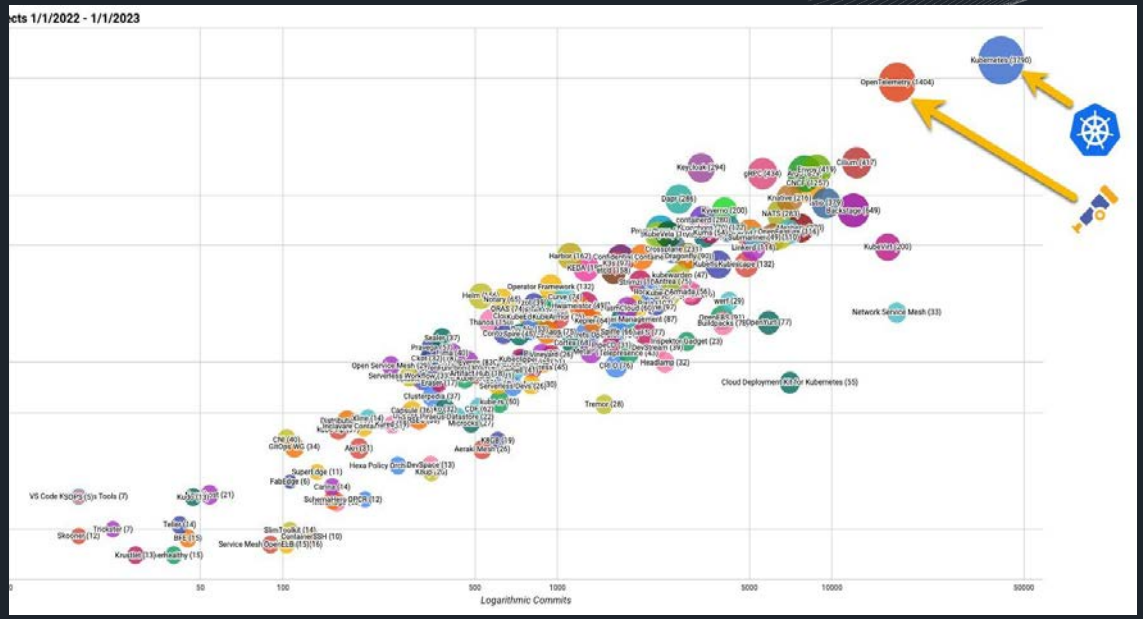
new relic

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.

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

new relic

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**

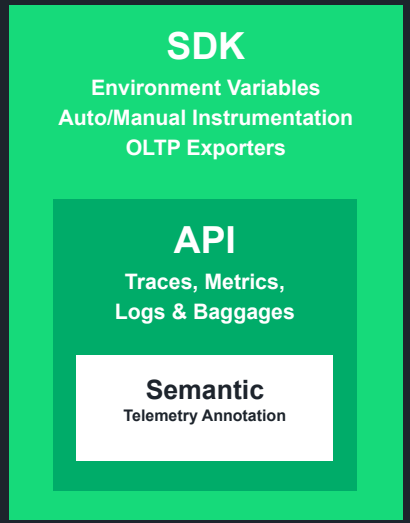
new relic

Core Concepts

The building blocks

new relic

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.

Application Instrumentation

Hands-On+Demo

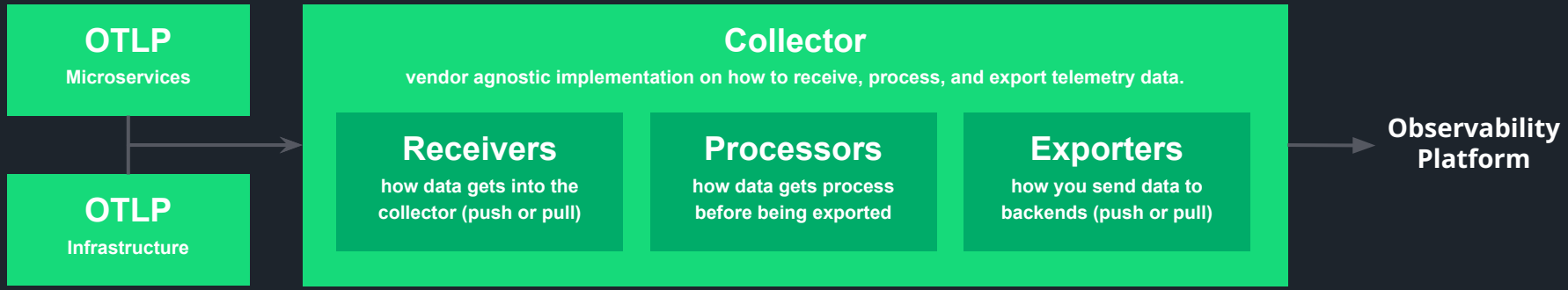
new relic

Infrastructure

The Collector

new relic

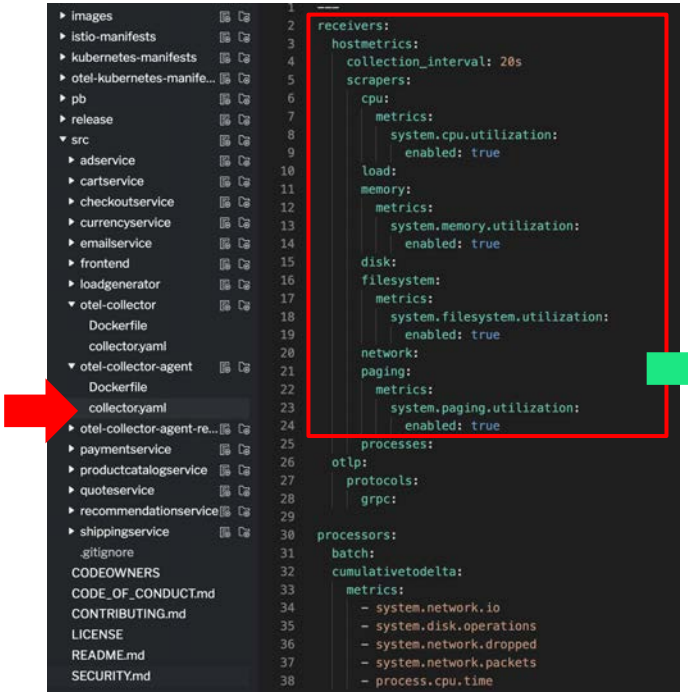
OpenTelemetry - Collector

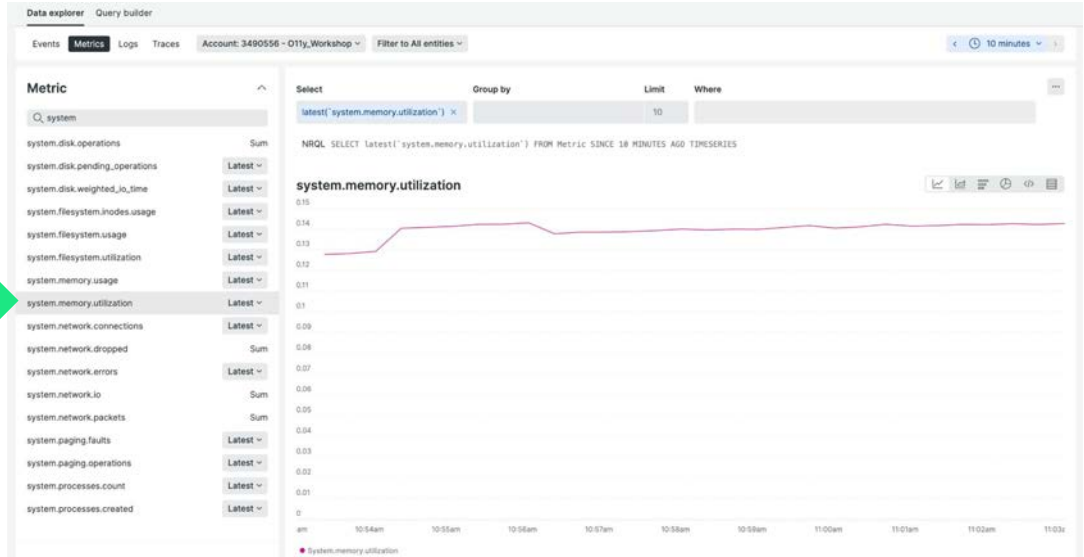


a **proxy** that process multiple telemetry formats, via an **agent** or a **gateway**
(e.g., OTLP, Jaeger, Prometheus, as well as many commercial/proprietary tools)

new relic

OpenTelemetry - Collector

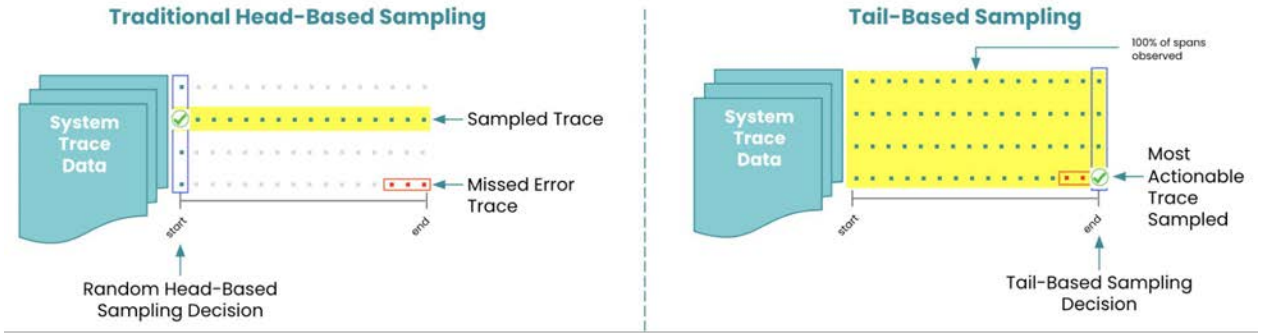




Collect additional infrastructure metrics for OpenTelemtry

OpenTelemetry - Collector

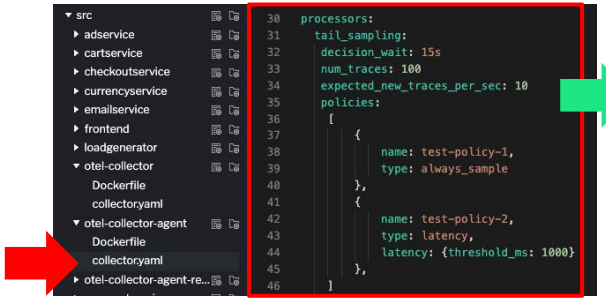
Storing all tracing data is **costly**. Most of the time, you **ONLY NEED THE RIGHT DATA**, when outages are occurring, or patterns are emerging.

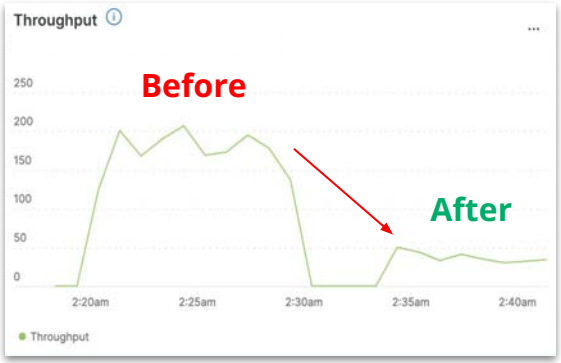


Head based sampling works well for an overall statistical sampling of requests through a distributed system.

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

OpenTelemetry - Collector

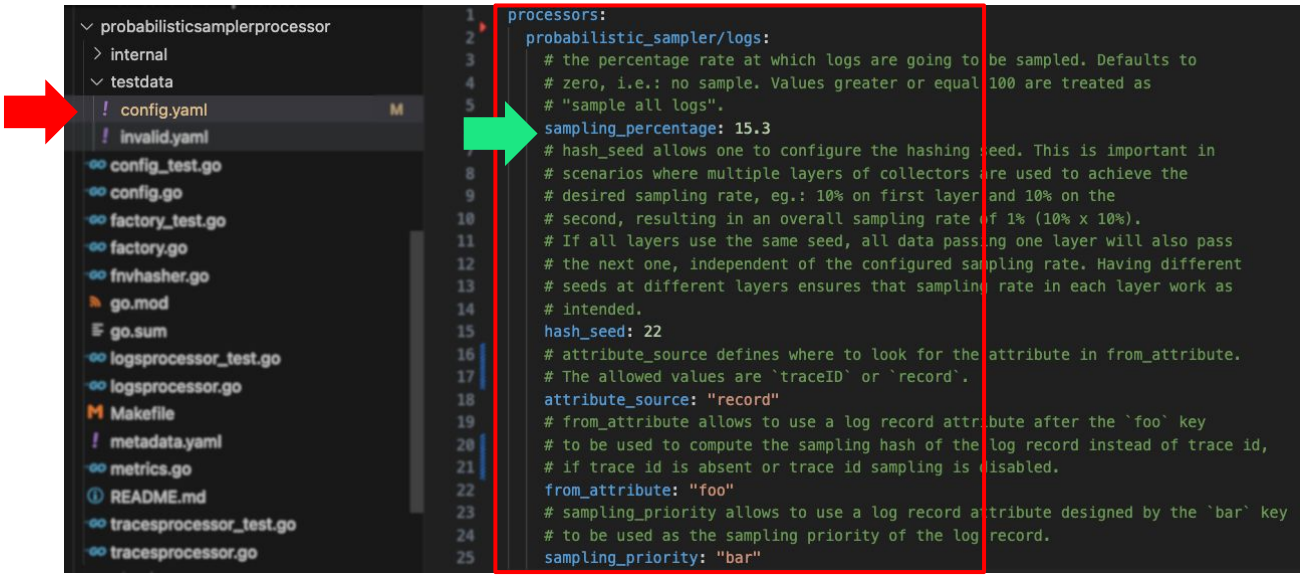




Effects from Tail Based Sampling

Apply processing decision in the Collector, before shipping data out

OpenTelemetry - Collector



Probabilistic Based Sampling

Severless

AWS Lambda

new relic

Lambda Instrumentation

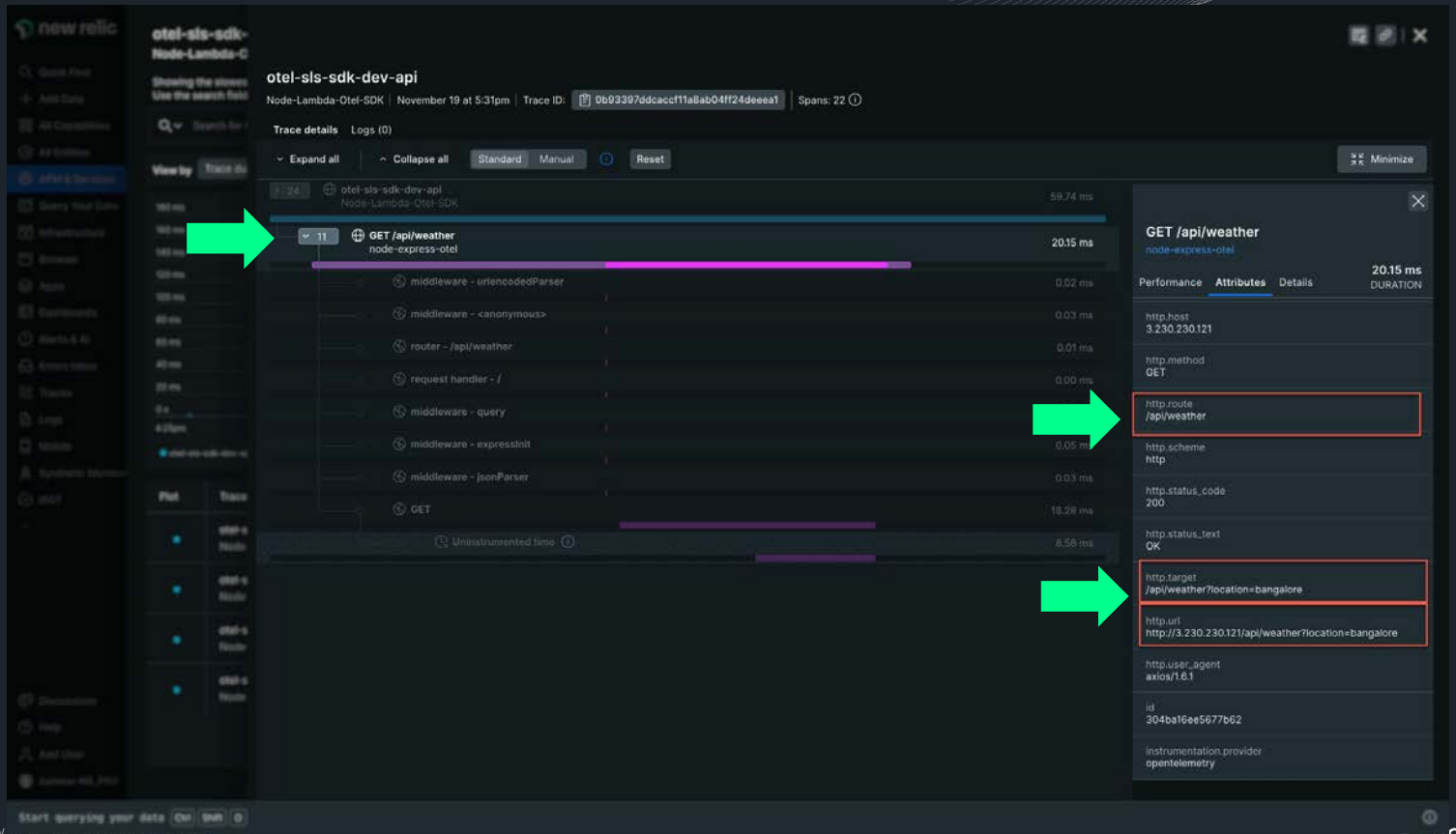
```
"@opentelemetry/instrumentation-aws-lambda": {
  enabled: true,
  disableAwsContextPropagation: true,
  requestHook: (span, { event, context }) => {
    span.setAttribute("faas.name", context.functionName);

    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"
```


new relic

Lambda Instrumentation



new relic

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"**.

new relic

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/

$whoami

Zameer Fouzan

Senior Developer Relations Engineer

new relic