OTEL ME ABOUT METRICS:

A Metrics 101 Crash Course

Reese Lee, New Relic

REESE LEE

Developer Relations Engineer

New Relic

- Previously Technical Support
- OpenTelemetry End User WG
  - Adoption and implementation
  - Feedback loop to improve the project
- Malaysia → Pacific Northwest
- The Netherlands is my 15th country!

AGENDA

01

METRICS OVERVIEW

02

OPENTELEMETRY OVERVIEW

03

METRICS DIP

04

WHAT'S NEXT

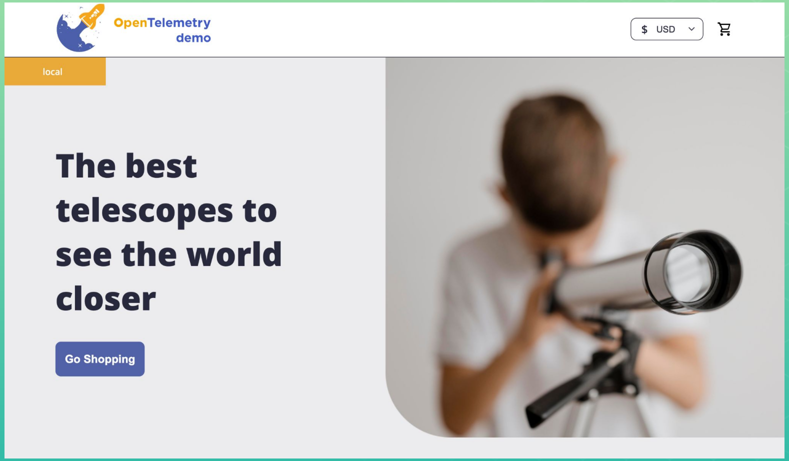
01

METRICS OVERVIEW

1. WHAT IS A METRIC?
2. WHY ARE METRICS USEFUL?

WHAT IS A METRIC?

A metric is a measurement about a service captured at runtime. Metrics represent aggregations of multiple measurements.



The best telescopes to see the world closer

- Throughput
- Response time
- Error rate
- CPU utilization
- Number of active users
- Total processed orders
- Total processed orders of a specific item

} **METRICS!**

WHY ARE METRICS USEFUL?

DATA VOLUME REDUCTION

Reducing the volume of data

PERFORMANCE

Monitoring your system

ALERTS

Alerting on breached SLOs

VISUALIZATION

Powering graphs and charts

02

OPENTELEMETRY OVERVIEW

WHAT IS OPENTELEMETRY?

WHAT IS OPENTELEMETRY?

OpenTelemetry is…

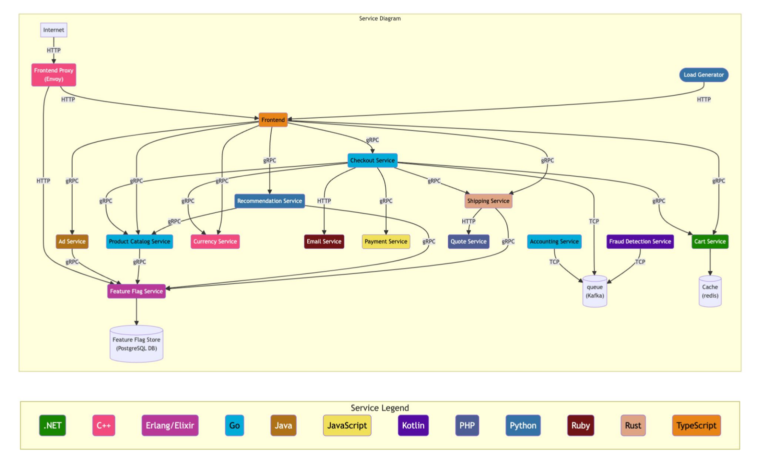
- An observability framework built on an open standard
- The merging of OpenCensus and OpenTracing in 2019
- 2nd most active CNCF project in terms of contributions (after Kubernetes)
- Aims to standardize instrumentation and telemetry generation, collection, and transmission

Considerations:

- OpenTelemetry is *not* a data visualization tool or storage solution

WHAT IS OPENTELEMETRY?

OpenTelemetry provides a set of APIs and SDKs, tools and components (such as the Collector), instrumentation libraries, semantic conventions,and a protocol (called OTLP).
Instrument once!



- Java
- .NET
- Python
- Ruby
- … and more

} ONE STANDARDIZED SET OF TOOLS

WHY OPENTELEMETRY FOR METRICS?

ABILITY TO CONNECT METRICS TO OTHER SIGNALS

Exemplars
Enrich metrics attributes via Baggage and Context

OPENCENSUS MIGRATION TO OPENTELEMETRY

Original goal of OpenTelemetry (OpenCensus + OpenTracing

WORKS WITH EXISTING METRICS INSTRUMENTATION PROTOCOLS

Minimum goal: Prometheus and Statsd

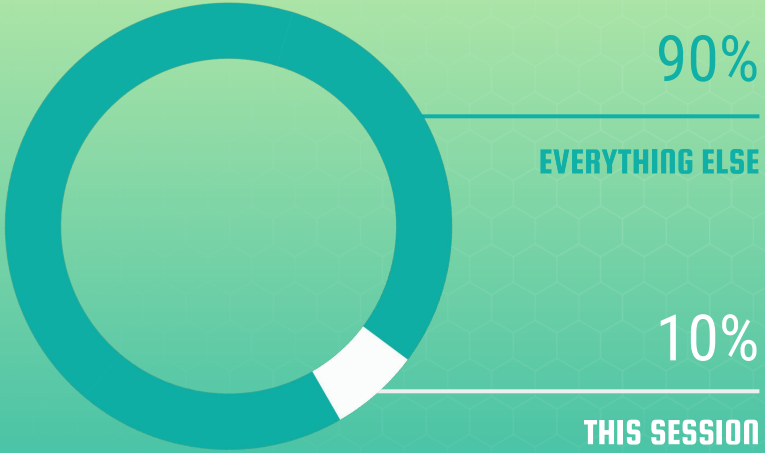
FREEDOM FROM VENDOR LOCK-IN

03

METRICS DIP

1. SESSION SCOPE
2. METRICS IN OPENTELEMETRY
3. ARCHITECTURE
4. METRIC INSTRUMENTS, TYPES, AND USE CASES
    a. What is an instrument?
    b. What instruments does OpenTelemetry provide?
    c. Why is instrument selection important?
    d. How do I choose an instrument?

SESSION SCOPE



So much more to get into! Deep dives, implementation, etc.

High level overview of select metrics concepts

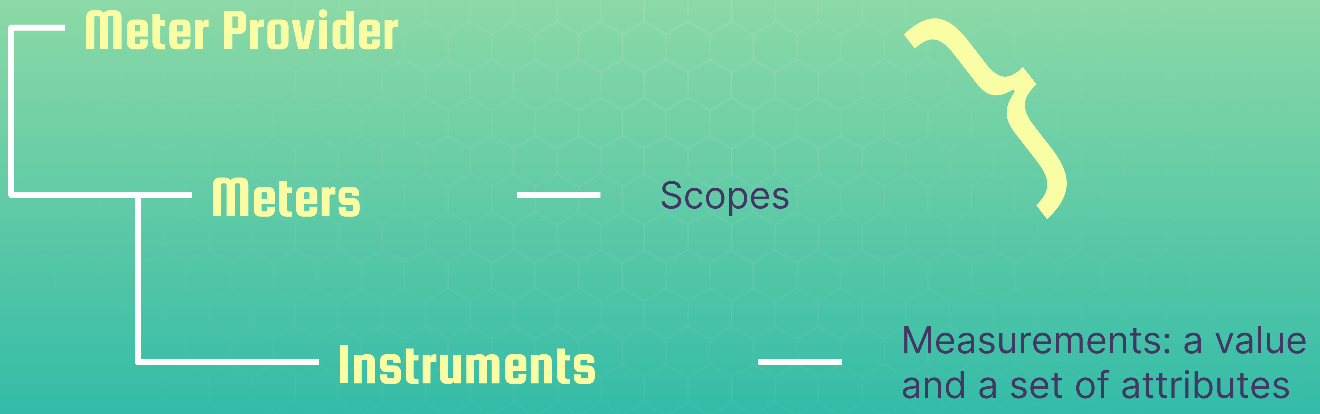
METRICS IN OPENTELEMETRY

API

Used to instrument code

SDK

Used to implement the API



Meter Provider

Meters — Scopes

Instruments — Measurements: a value and a set of attributes

METRICS IN OPENTELEMETRY

AGGREGATION

The process of combining multiple measurements into a single point

TEMPORALITY

Cumulative

Delta

Related to whether the reported values of additive quantities include previous measurements

Monotonic

Non-monotonic

MONOTONICITY

Related to whether the value is always increasing, or always increasing and decreasing at the same time

DIMENSIONS

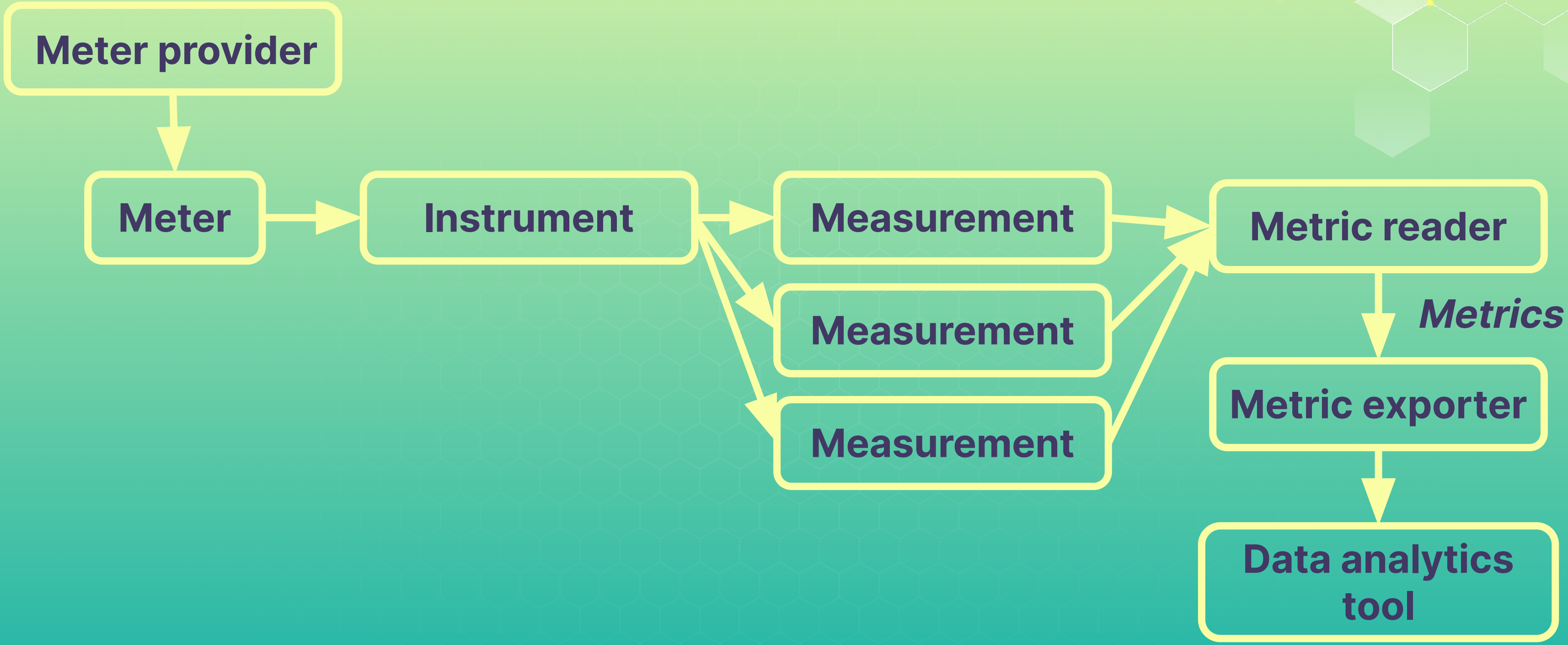
An attribute associated with a metric

CARDINALITY

How many unique dimensions are associated with a metric

Does It Add Up? Exploring the Delta Temporality in OpenTelemetry and Beyond - Matej Gera & Oded David, Coralogix

ARCHITECTURE



METRIC INSTRUMENTS, TYPES, AND USE CASES

What is an instrument?

Instruments report measurements and have the following fields:

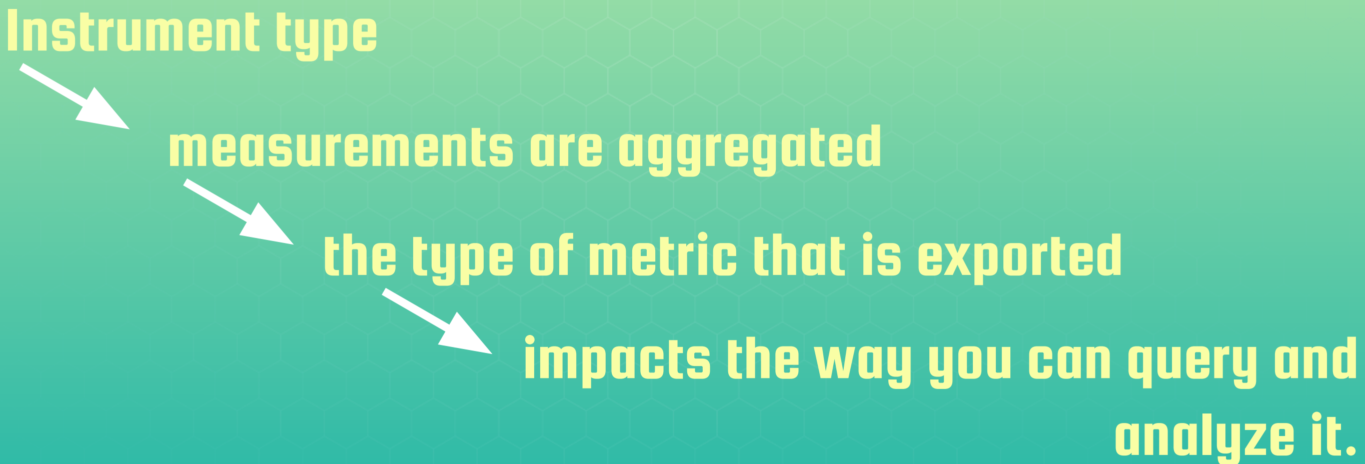
Instrument name	telescopes_sold
Kind	counter
Measure of unit (optional)	telescope
Description (optional)	“Total telescopes sold”

WHAT INSTRUMENTS DOES OPENTELEMETRY PROVIDE?

	INSTRUMENT	SYNCHRONOUS	ADDITIVE	MONOTONIC	AGGREGATION
1	Counter	■	■	■	Sum
2	Up/down counter	■	■	□	Sum
3	Async counter	□	■	■	Sum
4	Async up/down counter	□	■	□	Sum
5	Histogram	■	□	□	Histogram
6	Gauge	□	□	□	Last value

WHY IS INSTRUMENT SELECTION IMPORTANT?

Default aggregation reflects the intended use of the instrument



Instrument type

measurements are aggregated

the type of metric that is exported

impacts the way you can query and analyze it.

HOW DO I CHOOSE AN INSTRUMENT?

Analysis

How do you want to analyze the data?

Sync or async

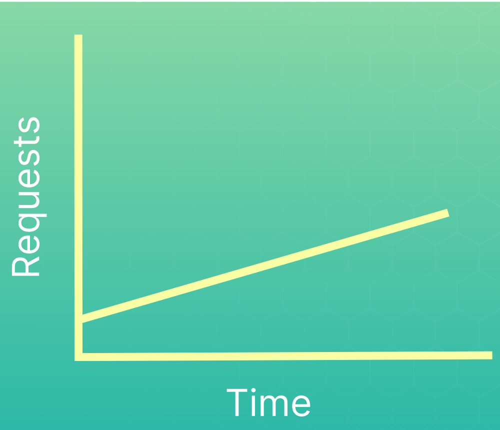
Do you need the measurement synchronously, or can it be reported on a set interval?

Monotonicity

Are the values monotonic?

COUNTER

Synchronous	Additive	Monotonic	Default aggregation	Example usage
✅	✅	✅	Sum	Number of bytes sent, total orders processed, total cart adds, total cart add failures, total checkouts, total checkout failures

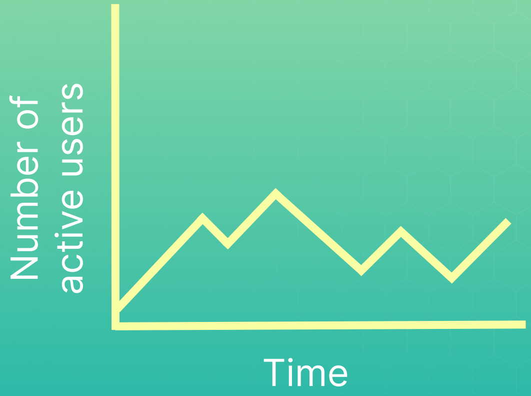


Use when...

- you want to count things and compute the rate at which things happen
- the sum of the things is more meaningful than the individual values

UP/DOWN COUNTER

Synchronous	Additive	Monotonic	Default aggregation	Example usage
✅	✅	❌	Sum	Number of open connections, number of active users, queue size, memory in use

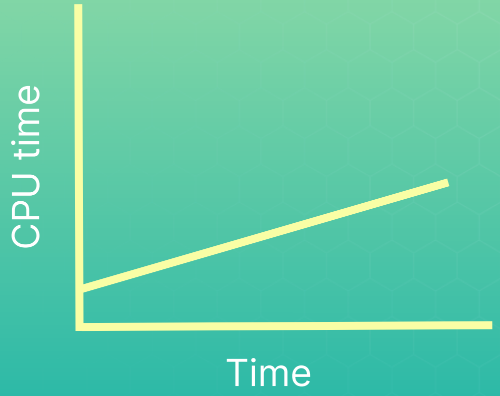


Use when…

- you have values that are negative, or that go up and down

ASYNC COUNTER

Synchronous	Additive	Monotonic	Default aggregation	Example usage
❌	✅	✅	Sum	CPU time, cache hits and misses, total network bytes transferred

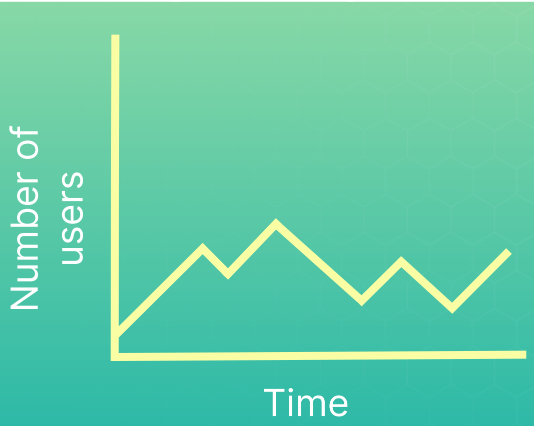


Use when…

- you need a sum of your measurements, but they may be too expensive to report synchronously, or it is more appropriate to record on set intervals

ASYNC UP/DOWN COUNTER

Synchronous	Additive	Monotonic	Default aggregation	Example usage
❌	✅	❌	Sum	Memory utilization, process heap size, number of active shards, changes in the number of active users

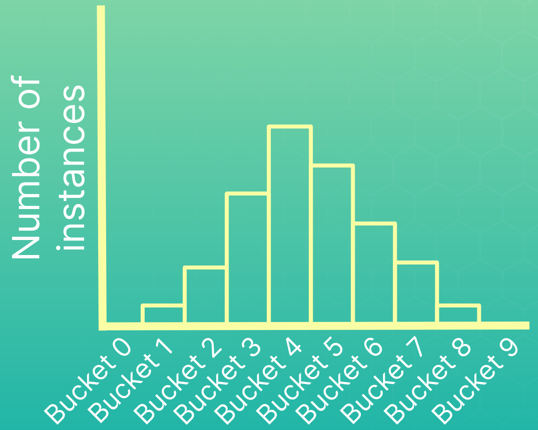


Use when...

- you need a non-monotonic additive counter to report on set intervals
- you need an absolute value, not a delta

HISTOGRAM

Synchronous	Additive	Monotonic	Default aggregation	Example usage
✅	❌	❌	Explicit bucket histogram	HTTP server response times, client duration, request rate



Buckets of HTTP server response times

Use when…

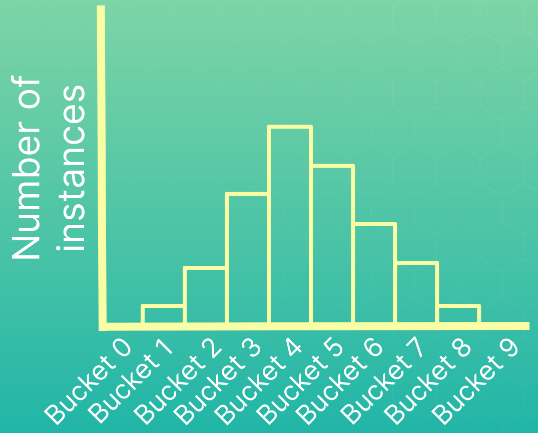
- you want to analyze the distribution of measurements to identify trends
- you want to calculate the min, max, and average response time

HISTOGRAM

OpenTelemetry also supports **exponential bucket histograms**! To learn more, check out:

Using OpenTelemetry's Exponential Histograms in Prometheus – Ruslan Kovalov & Ganesh Vernekar, Grafana Labs (talk)

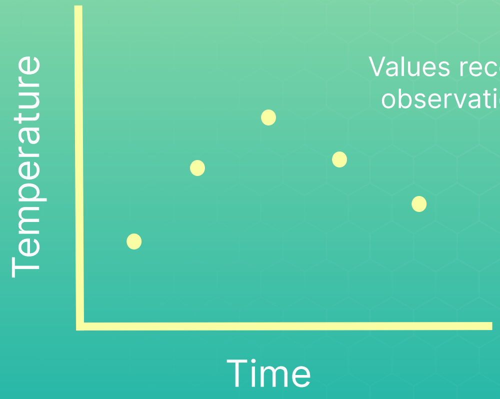
Exponential Histograms: Better Data, Zero Configuration – Jack Berg, New Relic (blog)



Buckets of HTTP server response times

GAUGE

Synchronous	Additive	Monotonic	Default aggregation	Example usage
❌	❌	❌	Last value	CPU utilization, temperature of hardware at this point in time, average memory consumption



Use when…

- you want to report data that’s not useful to aggregate across dimensions and you have access to measurements asynchronously
- you want finer-grain control of when a non-additive measurement is made, particularly when its purpose is a distribution

ASYNC UP/DOWN COUNTER vs GAUGE

ASYNC UP/DOWN COUNTER

- Non-monotonic
- Records an absolute value

GAUGE

- Records the last value

Depends on whether you need to sum values across dimensions.

- Use when you want to aggregate or sum across dimensions in a meaningful way
- Example: If you're tracking how many people visit a website, you might want to count how many visitors come from different countries or use different browsers, and then add all those counts together to get the total number of visitors.

- Use when you want to report data that's not useful to aggregate across dimensions, or when each individual measurement is important on its own and doesn't need to be summed together
- Example: It's not useful to sum across multiple distinct values of temperature readings

OPENTELEMETRY CONCEPTS

VIEWS

Allows you to customize the metrics output by the SDK:

- Process or ignore instruments
- Override aggregation strategy
- Attributes

04

WHAT'S NEXT

1. RECAP
2. WHAT TO EXPLORE NEXT?
3. CREDITS, REFERENCES & CONTACT INFO

RECAP

- What a metric is, and why they’re useful for observability
- What OpenTelemetry is, and some of the utility and customization options it provides in metric generation and collection
- Metrics concepts as they apply in OpenTelemetry
- OpenTelemetry metric instruments, and how to choose one

	Synchronous	Additive	Monotonic	Default aggregation	Example usage
Counter	✅	✅	✅	Sum	Number of bytes sent, total orders processed
Up down counter	✅	✅	❌	Sum	Number of open connections, number of active users
Histogram	✅	❌	❌	Histogram	Response times, search results latency
Async counter	❌	✅	✅	Sum	Cache hits and misses, CPU time
Async up down counter	❌	✅	❌	Sum	Memory utilization, number of active users
Gauge	❌	❌	❌	Last value	CPU utilization, hardware temperature

WHAT TO EXPLORE NEXT?

- Instrumentation and implementation - try it out yourself!
- Views API
- Data point types
- Adding metric attributes (or dimensions)
- Push- vs pull-based exporting
- Application runtime metrics
- OpenTelemetry collector metrics processors
- Infrastructure metrics
- ... and so much more!

CREDITS & REFERENCES

CREDITS

- Jack Berg, New Relic
- Vijay Samuel, eBay
- Alex Boten, Lightstep

REFERENCES

- Exponential Histograms: Better Data, Zero Configuration – Jack Berg
- Cloud-Native Observability with OpenTelemetry – Alex Boten
- OpenTelemetry docs
- OpenTelemetry Metrics Primer for Java Developers – Asaf Mesika
- Does It Add Up? Exploring the Delta Temporality in OpenTelemetry and Beyond - Matej Gera & Oded David, Coralogix
- Using OpenTelemetry's Exponential Histograms in Prometheus – Ruslan Kovalov & Ganesh Vernekar, Grafana Labs

THANK YOU!

@reesesbytes

Reese Lee (CNCF Slack)

This presentation template was created by Slidesgo, with icons and infographics by Flaticon