Contents

[Open Telemetry Building Blocks / 3 Pillars of Observability 1](#_Toc95914088)

[Run Jaeger Docker Image 1](#_Toc95914089)

[Run Zipkin 1](#_Toc95914090)

[Docker Image 1](#_Toc95914091)

[Run Zipkin Jar 1](#_Toc95914092)

[Run Zipkin Via Source Code 1](#_Toc95914093)

[Open Telemetry Defination 1](#_Toc95914094)

[Open Telemetry “Stack” 1](#_Toc95914095)

[SDK 1](#_Toc95914096)

[Instrumentation 2](#_Toc95914097)

[Processors 3](#_Toc95914098)

[Exporters 3](#_Toc95914099)

[Additional Data 4](#_Toc95914100)

[Resources & Detectors 4](#_Toc95914101)

[Sampler 5](#_Toc95914102)

[Collector 5](#_Toc95914103)

[A DB 5](#_Toc95914104)

[Visualization Layer 5](#_Toc95914105)

[Examples 5](#_Toc95914106)

[Span 5](#_Toc95914107)

# Open Telemetry Building Blocks / 3 Pillars of Observability

1. Logs: The application story.
2. Metrics: Numbers telling the statistical facts about the system
3. Trace: The context of why things are happening.
   1. Trace Events

# Run Jaeger Docker Image

docker run -d --name jaeger \

-e COLLECTOR\_ZIPKIN\_HOST\_PORT=:9411 \

-p 5775:5775/udp \

-p 6831:6831/udp \

-p 6832:6832/udp \

-p 5778:5778 \

-p 16686:16686 \

-p 14268:14268 \

-p 14250:14250 \

-p 9411:9411 \

jaegertracing/all-in-one:1.31

# Run Zipkin

## Docker Image

docker run -d -p 9411:9411 openzipkin/zipkin

## Run Zipkin Jar

curl -sSL https://zipkin.io/quickstart.sh | bash -s java -jar zipkin.jar

## Run Zipkin Via Source Code

* git clone <https://github.com/openzipkin/zipkin>
* cd zipkin
* ./mvnw -DskipTests --also-make -pl zipkin-server clean install
* java -jar ./zipkin-server/target/zipkin-server-\*exec.jar

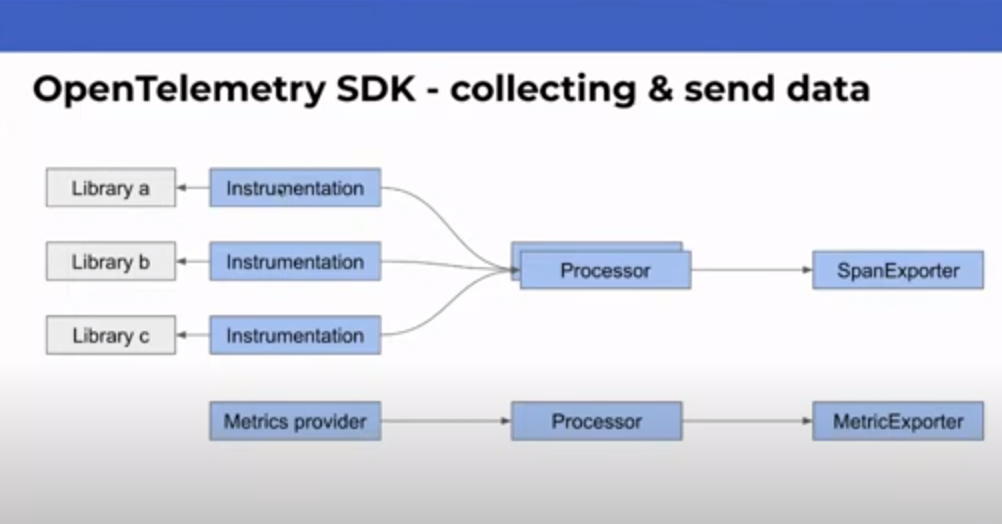
# Open Telemetry Defination

* The Glue to collect the three pillars together under a unified SDK
* Under CNCF (Cloud Native Compute Foundation)
* One Specification, Implementation for every programming language.

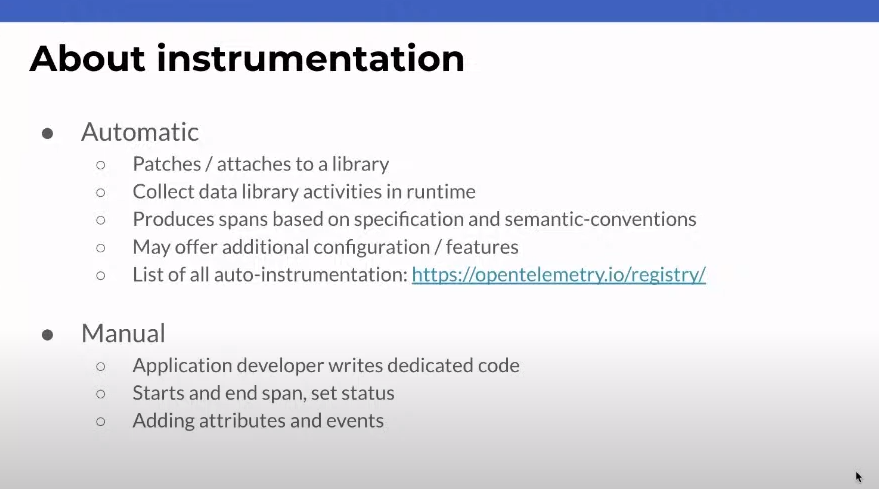
# Open Telemetry “Stack”

SDK:

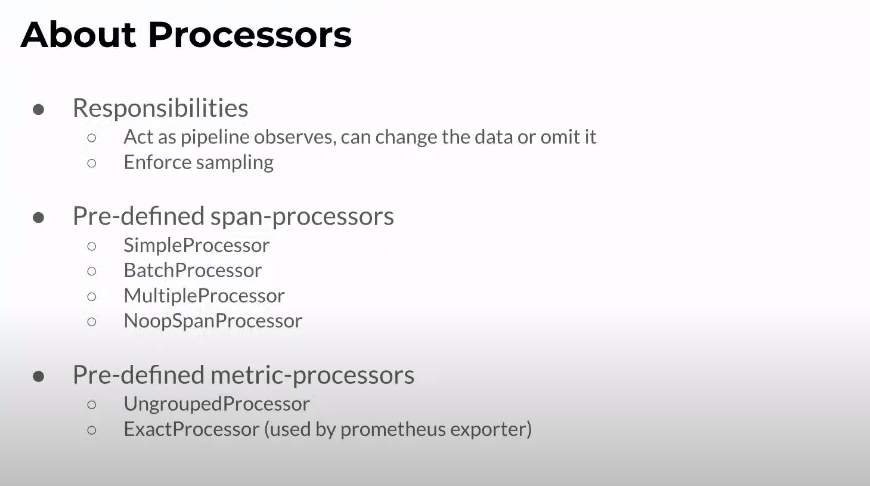
* Collect data about the application
* Propagate the context between the services
* Ship it somewhere.



### Instrumentation



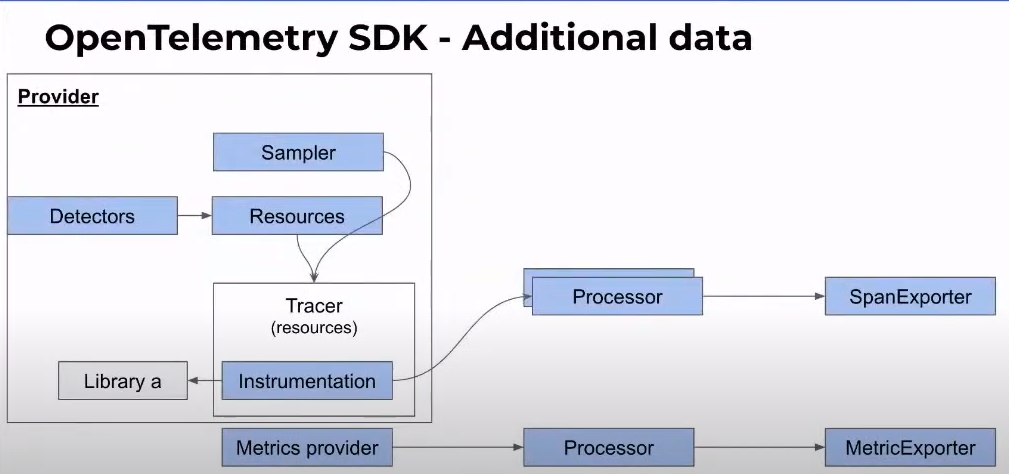
### Processors



### Exporters



### Additional Data



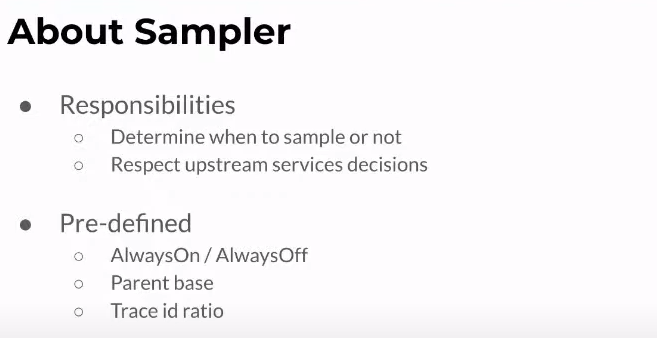
#### Resources & Detectors

* Send data not related to activity but data related to environment (i.e., metadata) example cloud region etc.

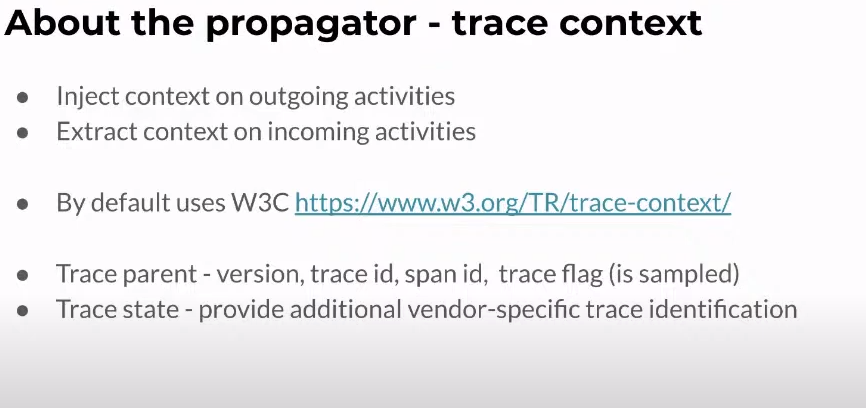


#### Sampler

You may want to sample the amount of data being sent out for various reasons.



### Context Propagation



Collector: receives telemetry, process it, and export it.

OpenSearch DB to store telemetry data

Visualization Layer.

# Examples

## Span

{

*traceId: '2f174037673c1c03f11de572f98d5adf',*

*parentId: undefined,*

*name: 'Refresh Cache',*

*id: '7d2665d949aa317b',*

*kind: 0,*

*timestamp: 1644994285835545,*

*duration: 1319594,*

*attributes: {},*

*status: { code: 0 },*

*events: []*

}