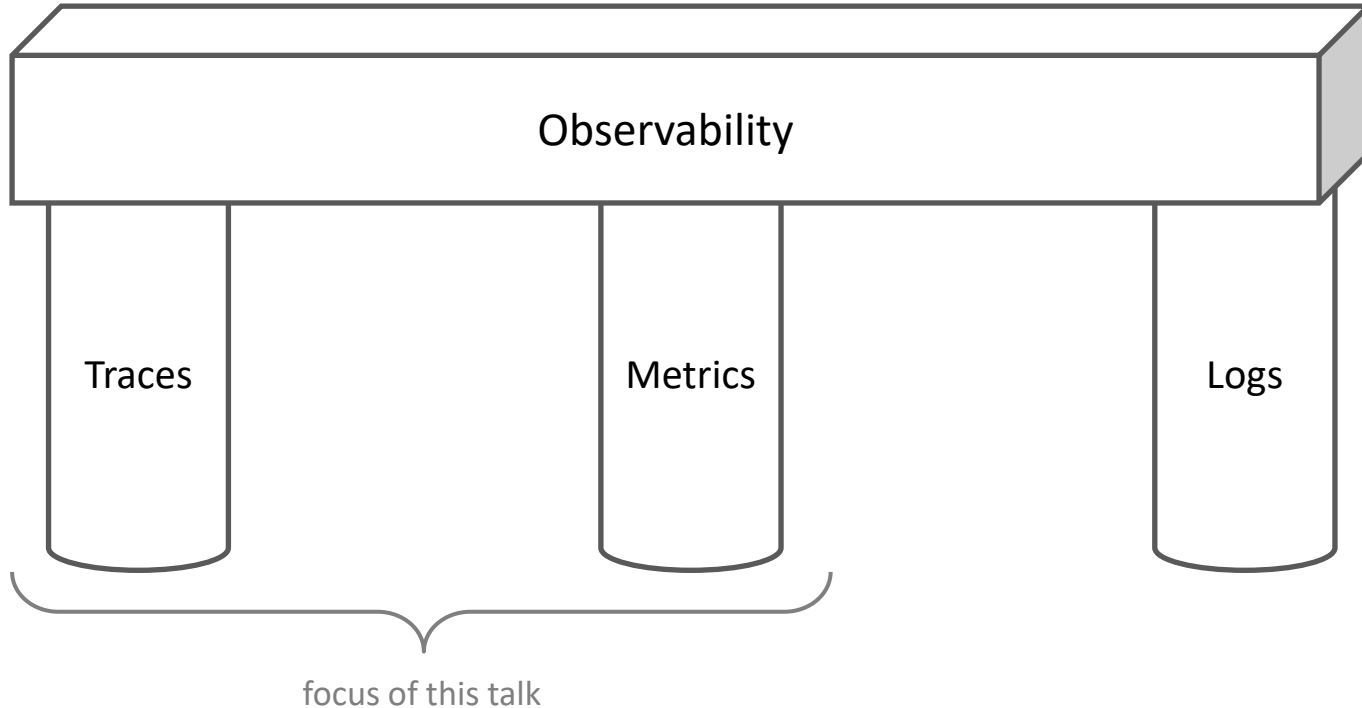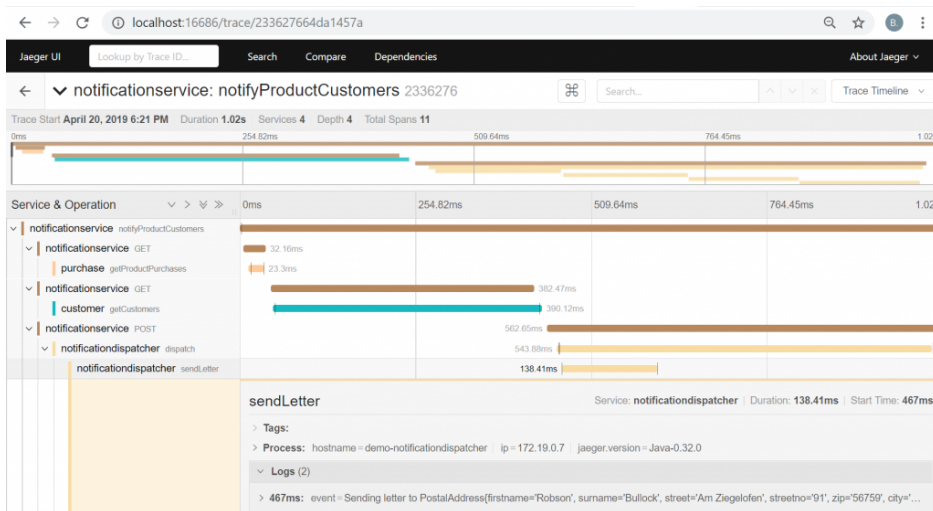# Observing Cloud-Native Java Apps
using OpenTelemetry
on
AWS, GCP and Azure

Bernhard Lubomski, RETIT GmbH, 2022/02/01

Three Pillars of Observability

Observability

Traces

Metrics

Logs

focus of this talk

Traces and Metrics

Jaeger

Source: https://grafana.com/products/cloud

Trace = directed, acyclic graph of spans

Span = represents unit of work. Properties:
- parent/child relation
- start time, end time
- tags, logs, errors…

Metric: time series of numbers to compute statistical values
- count
- average
- percentiles (median, 25th, 90th, …)
- …
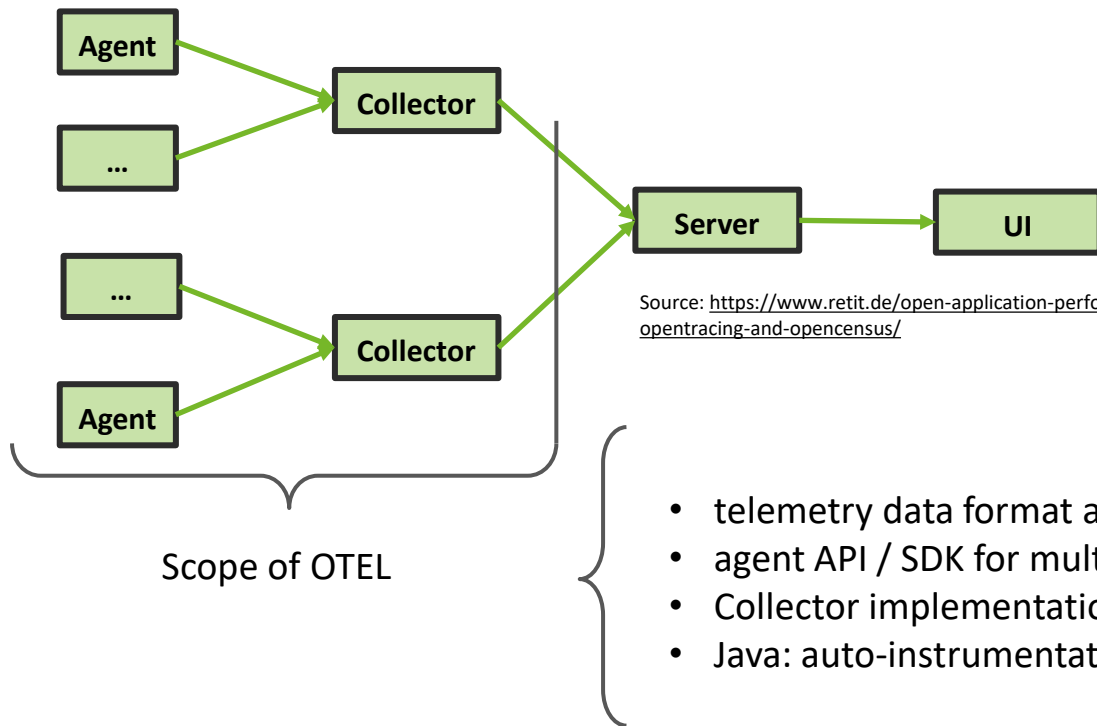
RETIT

OpenTelemetry (OTEL)

https://opentelemetry.io

- Observability Framework

- Collection of telemetry data: **traces**, **metrics, logs**

    - Defines **data formats** and **protocols** for traces, metrics and logs
    - Provides **APIs** for recording traces, metrics and logs
    - Provides **SDKs** with implementation of trace, metrics and logs* recording
    - Provides **Java auto-instrumentation** agent
      (https://github.com/open-telemetry/opentelemetry-java-instrumentation)

- Excludes: Tools for trace/metric/log storage, retrieval and visualization

*Logs: in draft stage (2022/01/24) https://opentelemetry.io/status/

Anatomy of an Observability Solution

Source: https://www.retit.de/open-application-performance-monitoring-apm-standards-opentracing-and-opencensus/

Scope of OTEL

- telemetry data format and protocols
- agent API / SDK for multiple languages
- Collector implementation
- Java: auto-instrumentation agent

RETIT

OpenTelemetry Java Auto-Instrumentation

Common libraries and frameworks are supported and calls are automatically traced.

Source 2022/01/27: https://github.com/open-telemetry/opentelemetry-java-instrumentation/blob/main/docs/supported-libraries.md

Akka Actors, Akka HTTP, Apache Axis2, Apache Camel, Apache CXF JAX-RS, Apache CXF JAX-RS Client
Apache Dubbo, Apache HttpAsyncClient, Apache HttpClient, Apache Kafka Producer/Consumer API
API, Apache MyFaces, Apache RocketMQ, Apache Struts 2, Apache Tapestry, Apache Wicket, Armer
Lambda, AWS SDK, Cassandra Driver, Couchbase Client, Dropwizard Views, Eclipse Grizzly, Eclipse Je
Client, Eclipse Metro, Eclipse Mojarra, Elasticsearch API, Elasticsearch REST Client, Finatra, Geode C
Grails, gRPC, Guava ListenableFuture, GWT, Hibernate, HttpURLConnection, http4k †, Hystrix, Java E
java.util.logging, JAX-RS, JAX-RS Client, JAX-WS, JDBC, Jedis, JMS, JSP, Kotlin Coroutines, Kubernetes
Log4j 2, Logback, Micrometer, MongoDB Driver, Netty, OkHttp, Play, Play WS, Quartz, RabbitMQ Clie
Reactor Netty, Rediscala, Redisson, RESTEasy, Restlet, RMI, RxJava, Scala ForkJoinPool, Servlet, Spark
Batch, Spring Data, Spring Integration, Spring Kafka, Spring RabbitMQ, Spring Scheduling, Spring We
Services, Spring WebFlux, Spymemcached, Twilio, Undertow, Vaadin, Vert.x Web, Vert.x HttpClient, V

Spring-*
JDBC, Hibernate
HTTP-Clients
JAX-WS, JAX-RS
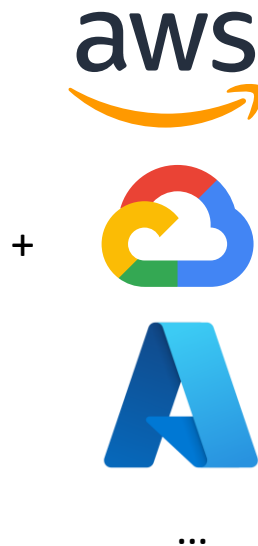**JMS**
Loggers
...

Application Server:
Jetty 9.4.x, 10.0.x, 11.0.x, Payara 5.0.x, 5.1.x, Tomcat 7.0.x, Tomcat 7.0.x, 8.5.x, 9.0.x, 10.0.x, TomEE 7
Liberty Profile 20.x, 21.x, Websphere Traditional 8.5.5.x, 9.0.x, WildFly 13.x, WildFly 17.x, 21.x, 25.x
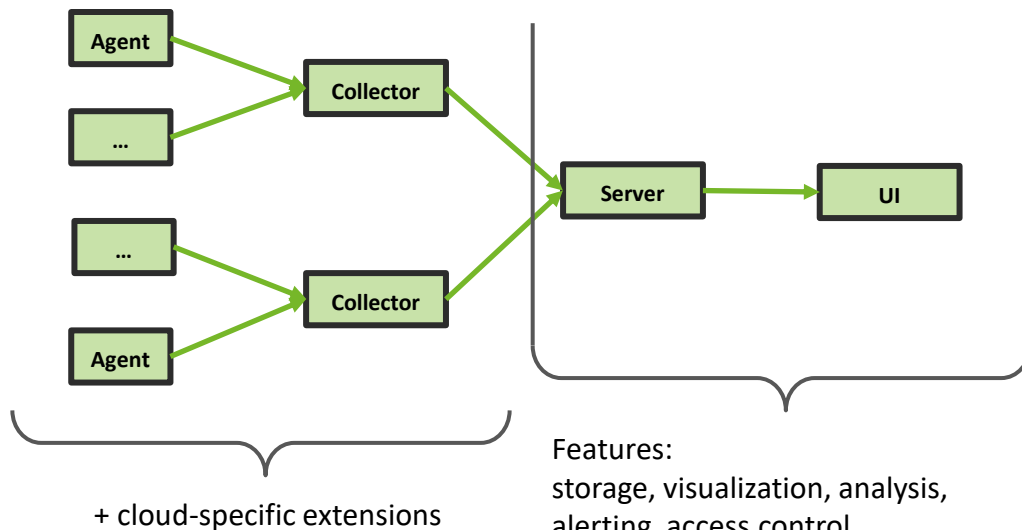
Typical OTEL Setup for Java
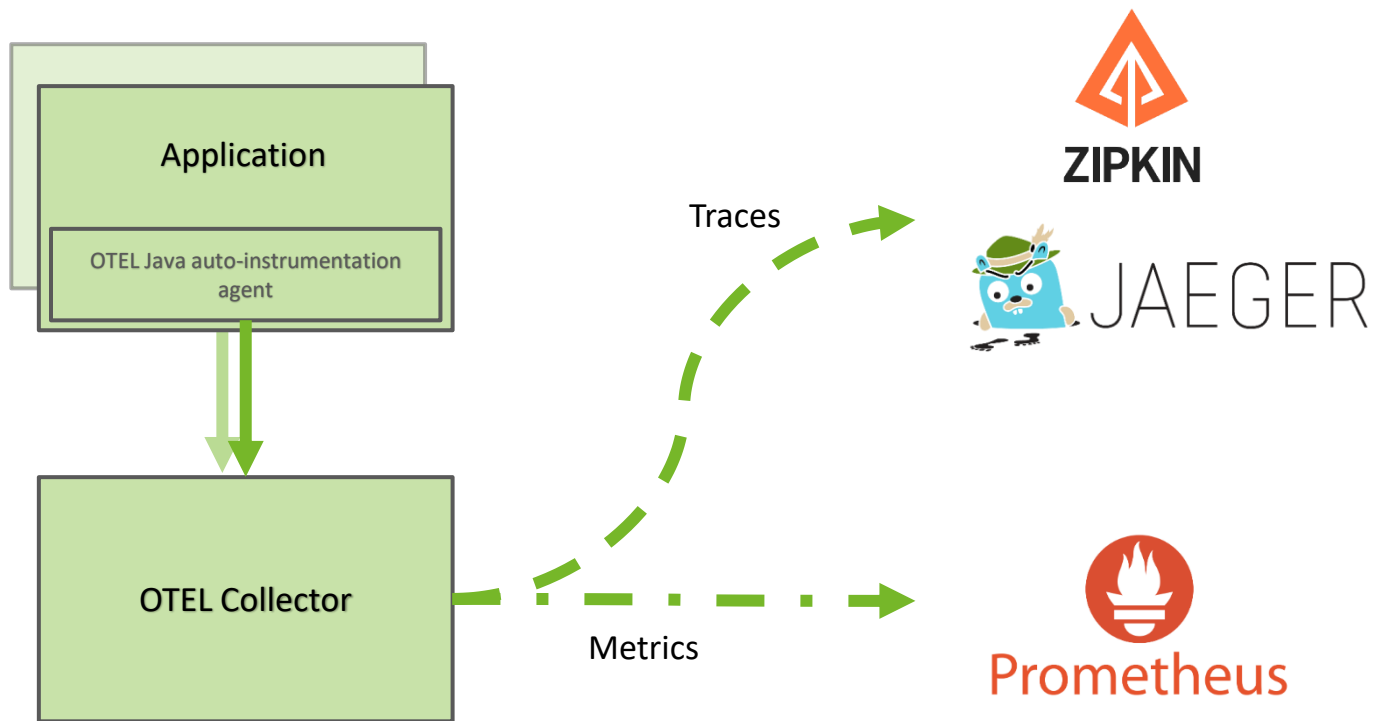
OpenTelemetry: Supported by Cloud Providers

- How is OTEL integrated and supported in AWS, GCP and Azure?

- Which are the benefits of the integration?

- How to make applications observable with OTEL in these cloud environments?

OTEL Integration by Cloud Providers

+ cloud-specific extensions

Features:
storage, visualization, analysis,
alerting, access control…

Managed Trace & Metrics Services

OpenTelemetry support: use managed trace and metrics services through OpenTelemetry + extensions. Normally, **usage of vendor proprietary SDKs** to record traces and metrics is required.

OTEL benefits over vendor proprietary SDKs:
- Java auto-instrumentation
- Vendor neutral OTEL API for (manually) recording traces and metrics

	Trace Service	**Metrics Service**
AWS	X-Ray	CloudWatch
GCP	Cloud Trace (formerly "Stackdriver")	Cloud Monitoring Metrics (formerly "Stackdriver")
Azure	Azure Monitor Application Insights	Azure Monitor Metrics (Not supported through OTEL API, yet)

Architectural Blueprint for OTEL integration by Cloud Providers

OTEL in AWS
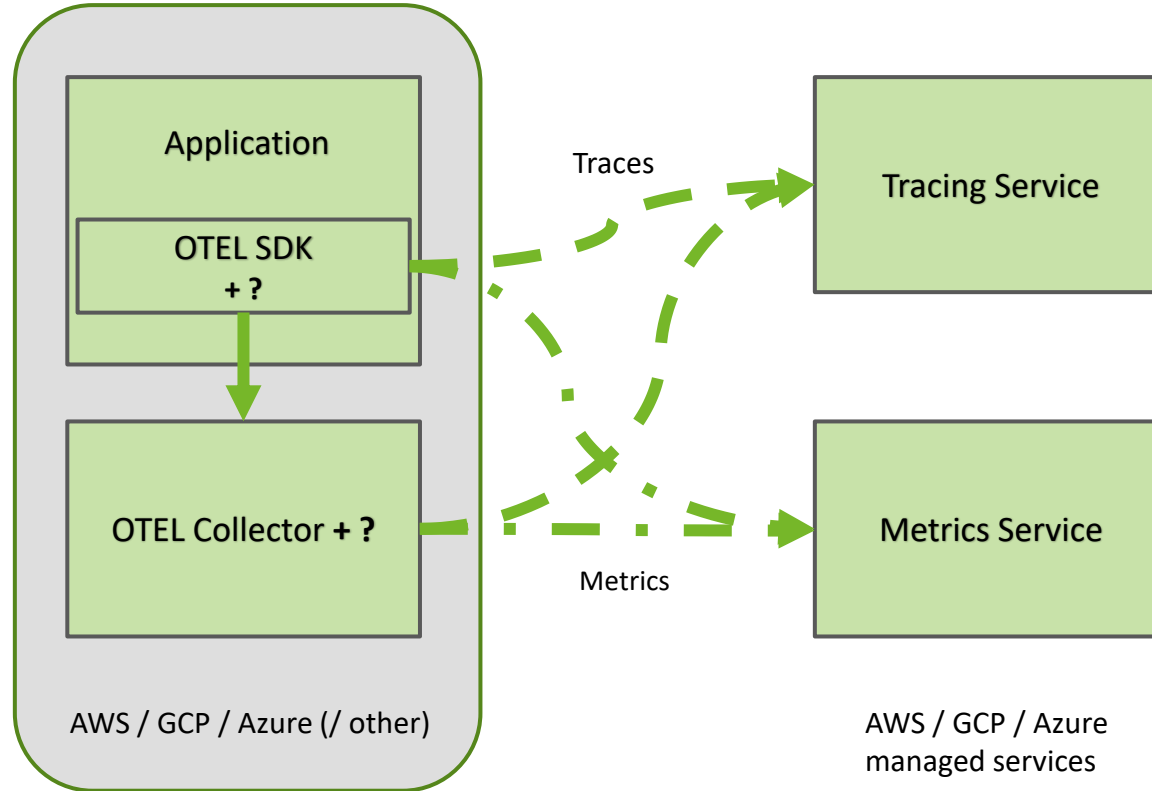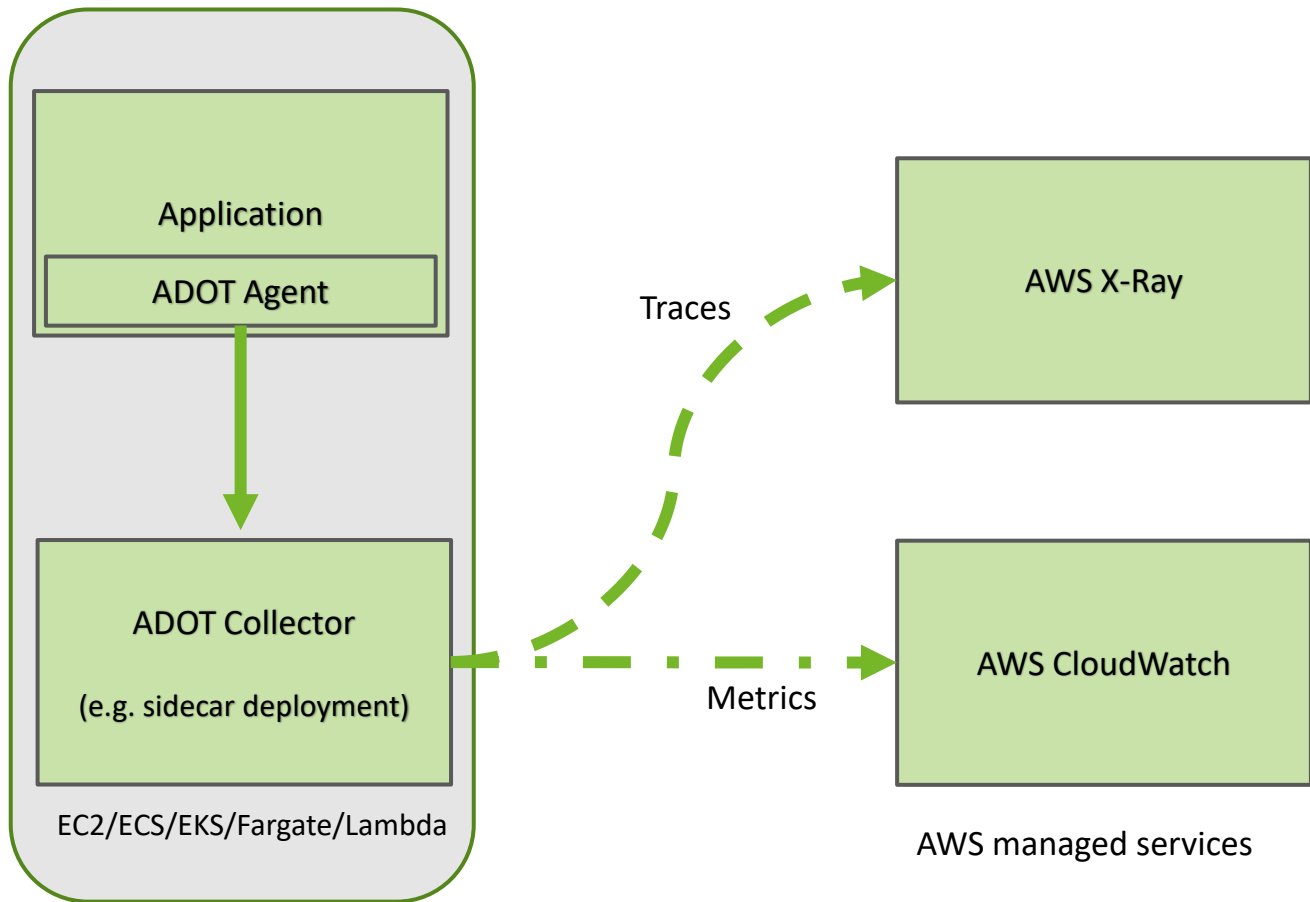
AWS Distribution for OpenTelemetry (ADOT) consists of:

- Java-auto-instrumentation agent redistribution with AWS pre-configuration (ADOT Agent)
  https://github.com/aws-observability/aws-otel-java-instrumentation

- Collector redistribution with AWS exporters for sending telemetry to X-Ray and CloudWatch (ADOT Collector)
  https://github.com/aws-observability/aws-otel-collector
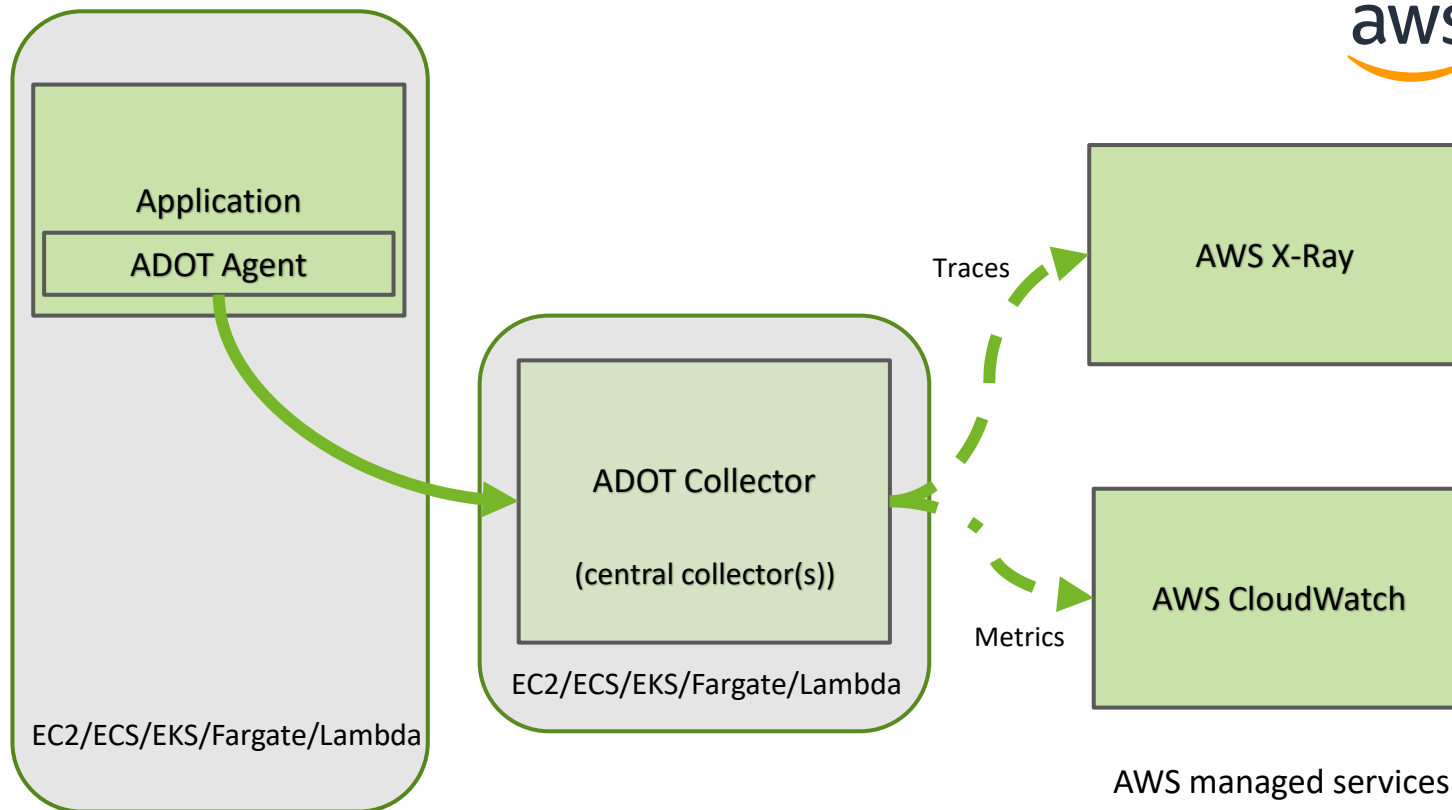
AWS OTEL Integration

Collector Sidecar

Application

ADOT Agent

ADOT Collector

(e.g. sidecar deployment)

EC2/ECS/EKS/Fargate/Lambda

Traces

Metrics

AWS X-Ray

AWS CloudWatch

AWS managed services

aws

RETIT

AWS OTEL Integration

Collector Service

AWS

Application

ADOT Agent

EC2/ECS/EKS/Fargate/Lambda

ADOT Collector

(central collector(s))

EC2/ECS/EKS/Fargate/Lambda

Traces

AWS X-Ray

Metrics

AWS CloudWatch

AWS managed services

AWS X-Ray

- Trace storage
- Trace exploration
- Statistics on traces
- Service Map

AWS CloudWatch

- Metrics storage
- Metrics visualization
- Alerts
- …

X-Ray console shown on last slide was recently integrated into CloudWatch view

Source https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/graph_a_metric.html

RETIT

OTEL in GCP

- Relies on OTEL API/SDK and code to manually record traces and metrics.
- Transmission of telemetry to GCP via extension of OTEL SDK:
  - https://github.com/GoogleCloudPlatform/opentelemetry-operations-java/tree/main/exporters/trace
  - https://github.com/GoogleCloudPlatform/opentelemetry-operations-java/tree/main/exporters/metrics

- **Traces** support: stable
- **Metrics** support: still based on OTEL Alpha SDK for metrics (2022/01/26)

- **OTEL auto-instrumentation** support: Exists, but based on still unstable custom exporter feature in OTEL, therefor considered "proof-of-concept" (2022/01/26)

GCP OTEL Integration

Application

OTEL metrics collection code

OTEL trace collection code

OTEL API/SDK

Google Cloud Trace Exporter

Google Cloud Monitoring Exporter

GCP / on-premise / other Cloud Provider

Traces

Metrics

Cloud Trace

Cloud Monitoring Metrics

managed services of the GCP Operations Suite (formerly Stackdriver)
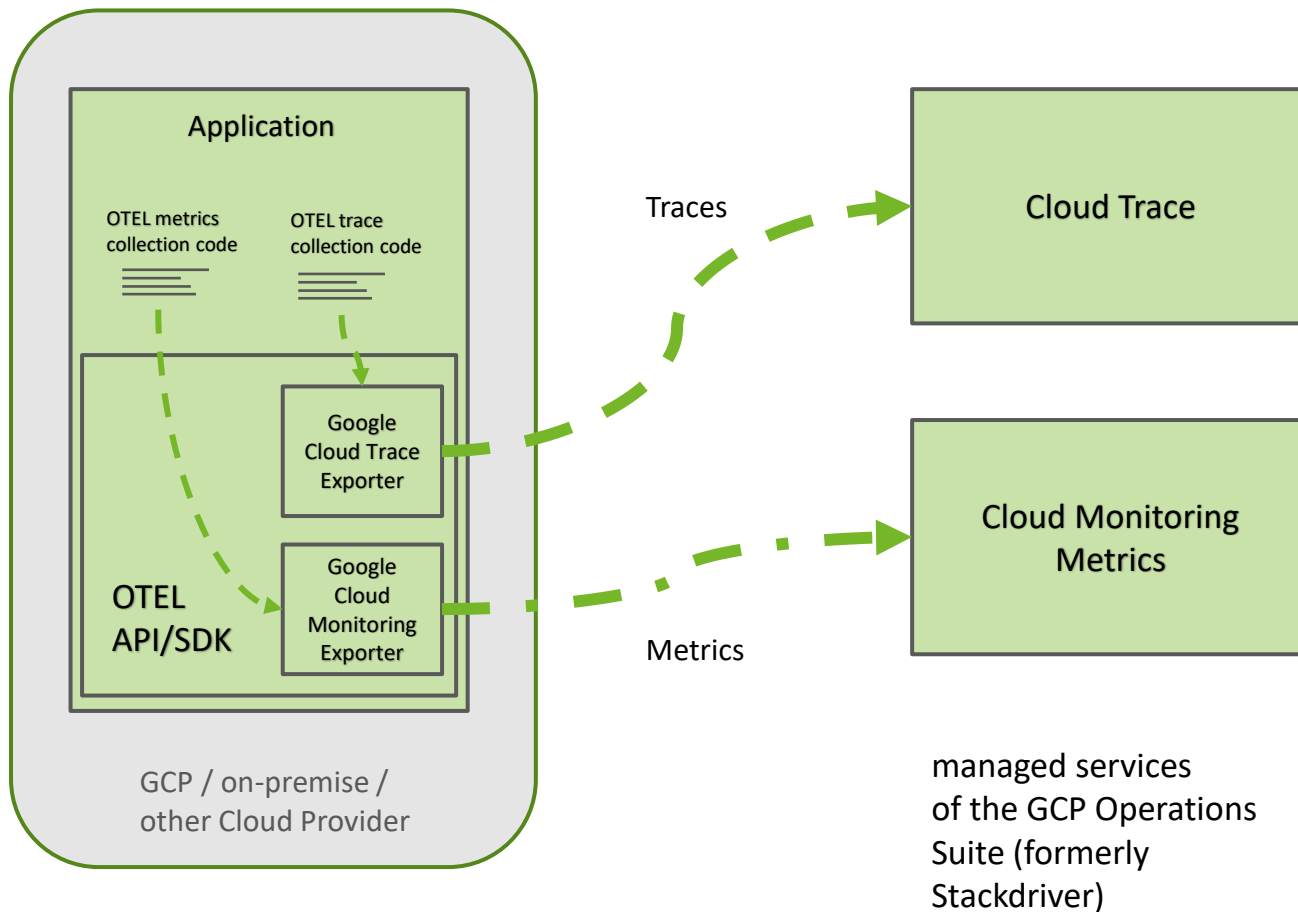
Workaround for recording Traces using Java- Auto-instrumentation in GCP

managed services of the GCP Operations suite

GCP auto-instrumentation exporter considered proof-of-concept, but Zipkin trace exporter is integrated into the base OTEL auto-instrumentation client.

Application

OTEL Java auto-instrumentation agent

Zipkin exporter

GCP / on-premise / Other Cloud Provider

Traces

(Zipkin format)

Cloud Trace

Cloud Trace Zipkin Collector
https://github.com/openzipkin/zipkin-gcp

GCP / on-premise / Other Cloud Provider

RETIT
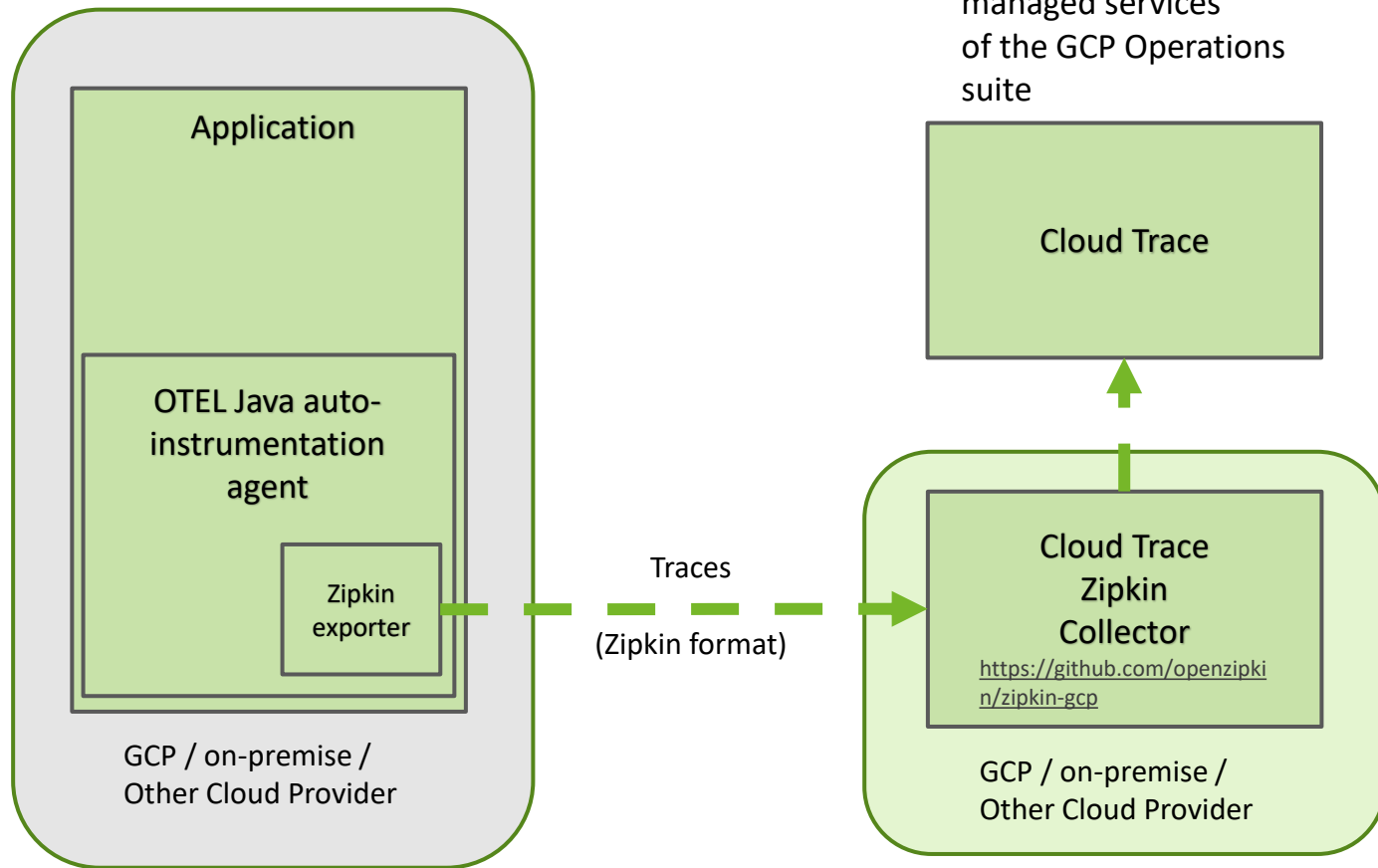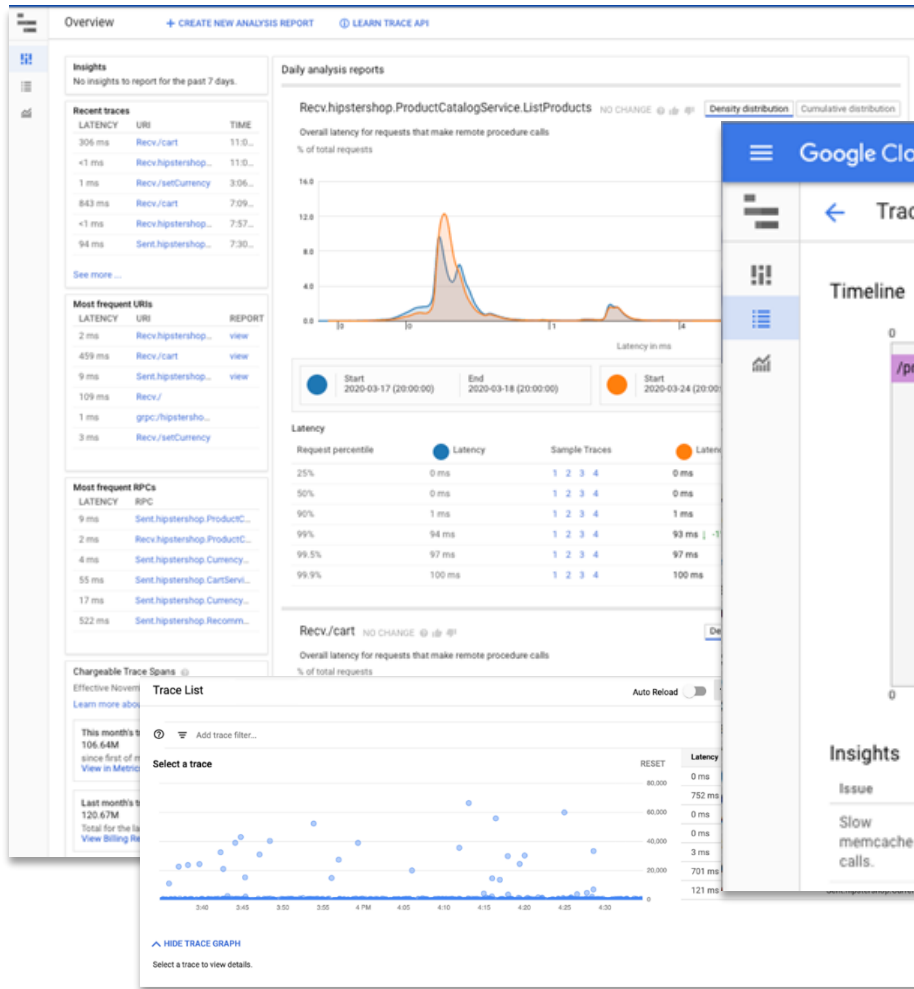
GCP Cloud Trace

- Trace storage
- Trace visualization
- Statistics on traces
- Suggestions on recognized bottlenecks

GCP
Cloud Monitoring
Metrics

- Metrics storage
- Metrics visualization
- Alerts
- ...

OTEL in Azure

- Java support based on OTEL auto-instrumentation, GA'd 2020
  - https://github.com/microsoft/ApplicationInsights-Java
- Direct transmission of traces to Azure application insights.

- <u>No</u> support for custom traces and metrics captured through OTEL API, *yet*.
  - Traces: application-insights.2.x SDK API
  - Metrics: Micrometer API, application-insights.2.x SDK API
    (https://docs.microsoft.com/en-us/azure/azure-monitor/app/java-in-process-agent#send-custom-telemetry-by-using-the-2x-sdk)

Azure OTEL Integration

Application

Azure Monitor Auto-Instrumentation for Java Agent
(based on OTEL + Azure SDK + Micrometer + …)

on-premises / *-cloud

Traces

Metrics

Azure Monitor Application Insights

(Traces + Metrics)

Azure managed services

fabrikamprod - Search
Application Insights

Search (Ctrl+/)

Tags

✕ Diagnose and solve problems

Investigate

Application map

Smart Detection

Live Metrics Stream

Metrics

Search

Availability

Failures

Performance

Servers

Browser

Troubleshooting guides (pre...

Workbooks

Usage

Users

🕐 Time range   ▼ Filters   ↻ Refresh   ↩ Reset   ▦ Analytics   ☺ Feedback   ⬈ Help

⚠ Data received from your application is being sampled to reduce the volume of telemetry data retained; Application Insights. Click to learn more.

Search

Event Types

Trace ✕   Page View

Property Filters

⊕

188 K total results betwee...

6K
4K
2K
0K
▲

Trace	Dependency
84 K	**55 K**

Results   ↻ Grouped res...

▌ 3/14/2019, 7:06:41 AM -
New Request Received
Severity level: Verbose

▌ 3/14/2019, 7:06:41 AM -
New Request Received
Severity level: Verbose

🔍 Search results   📖 Learn more

End-to-end transaction
Operation ID: b8e278fcfd20456a88ab48ac9cde6476 ▦

🟦 = Request (incoming)   ▨ = Dependency (outgoing)   ⚠ = Exception   🖼 = Profiler trace

EVENT		RES.	DURATION	0MS 200 MS 300 MS 400 MS 500 MS
⌄ 🔴 **App Service Customer Details** Central US			557 ms	▬▬▬▬
⌄ 🔵 **http://fabrikamfiberapp.azurewebsites.net/Customers/Details/8469**			557 ms	▬▬▬▬
⌄ 🔴 fabrikamfiberapp GET Customers/Details		500	430.3 ms	▬▬▬
🟦 fabrikamxyz FabrikamSQL			57 ms	▨
🟥 fabrikamaccount POST fabrikamaccount/Tables		409	112 ms	▨
🟥 fabrikamaccount POST fabrikamaccount/fabrikamfiber		409	111 ms	
🟥 fabrikamaccount GET fabrikamaccount/fabrikamfiber		200	48 ms	
⚠ **EXCEPTION** System.FormatException				🖼
🔵 **Failed** Central US			0	

Events   **View all telemetry** ⌃

fabrikamfiberapp
GET Customers/Details

☺ Fee

azurewebsites.net/Cust

Call Stack   ☐ Show Just

System.FormatException
    at System.Number.StringToNumber (mscorlib, Version=4.0.0...
    at System.Number.ParseInt32 (mscorlib, Version=4.0.0.0, C...
    at System.Int32.Parse (mscorlib, Version=4.0.0.0, Culture...
    at FabrikamFiber.DAL.Data.AddressValidator.ValidZipCode (...
    at FabrikamFiber.DAL.Models.Address.FullAddress (Fabrikam...
    at ASP._Page_Views_Customers_Details_cshtml.Execute (App_...
    at System.Web.WebPages.WebPageBase.ExecutePageHierarchy (...
    at System.Web.Mvc.WebViewPage.ExecutePageHierarchy (Syste...
    at System.Web.WebPages.StartPage.RunPage (System.Web.WebP...
    at System.Web.WebPages.StartPage.ExecutePageHierarchy (Sy...
    at System.Web.WebPages.WebPageBase.ExecutePageHierarchy (...
    at System.Web.Mvc.RazorView.RenderView (System.Web.Mvc, V...
    at System.Web.Mvc.BuildManagerCompiledView.Render (System...
    at System.Web.Mvc.ViewResultBase.ExecuteResult (System.We...
    at System.Web.Mvc.ControllerActionInvoker.InvokeActionResu...

Azure Monitor Application Insights (Transactions)

- Trace storage
- Trace visualization
- Profiling
- …

Source:
https://docs.microsoft.com/en-us/azure/azure-monitor/app/java-2x-trace-logs
https://docs.microsoft.com/en-us/azure/azure-monitor/app/transaction-diagnostics

Sum Server requests, Avg Server response time, and Sum Failed requests for tailspintoyswebapp ✏️

Add metric | Add filter | Apply splitting | Line chart ∨ | Drill in | ⋯

💡 tailspintoyswebapp, **Server requests, Sum** ✕ | 💡 tailspintoyswebapp, **Server response time,** Avg ✕ | 💡 tailspintoyswebapp, **Failed req**
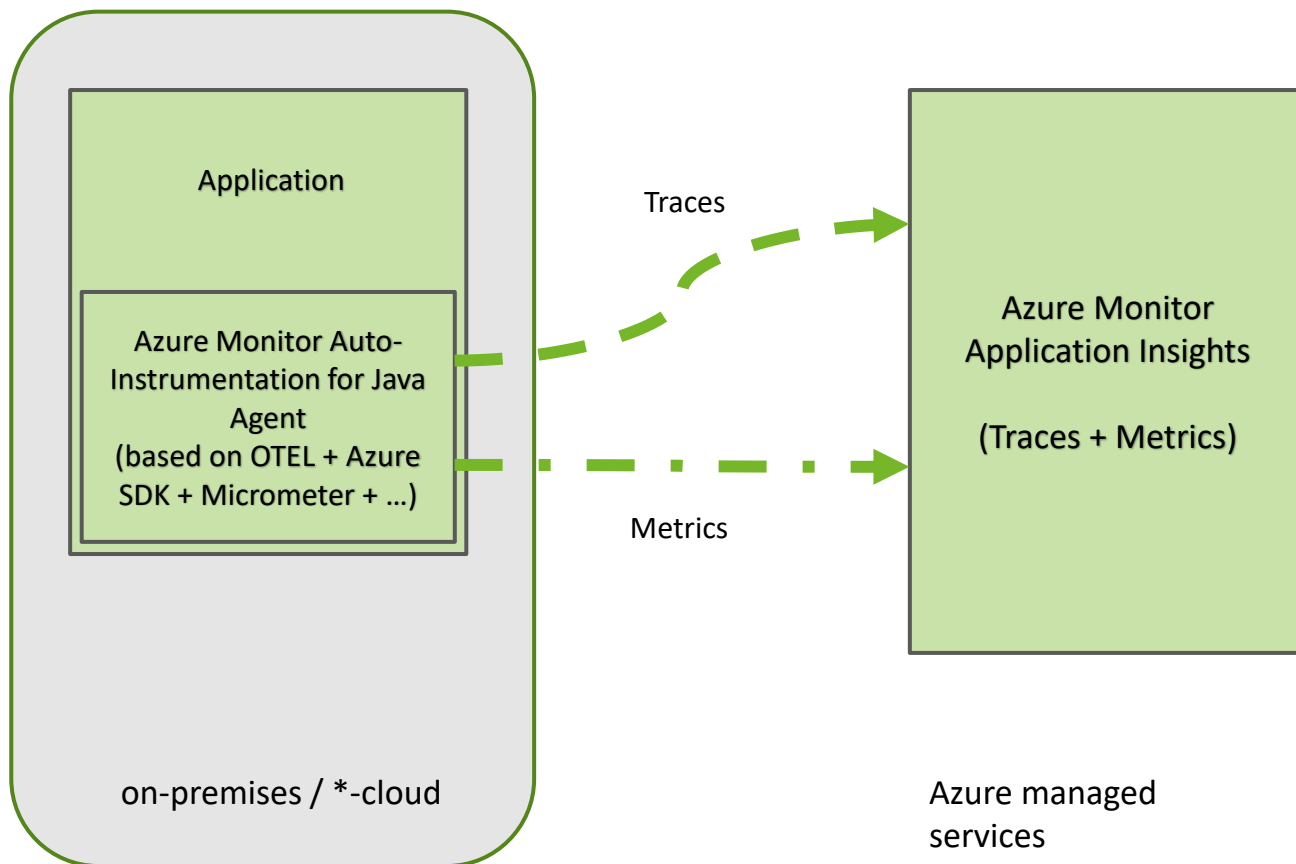
Azure Monitor Metrics
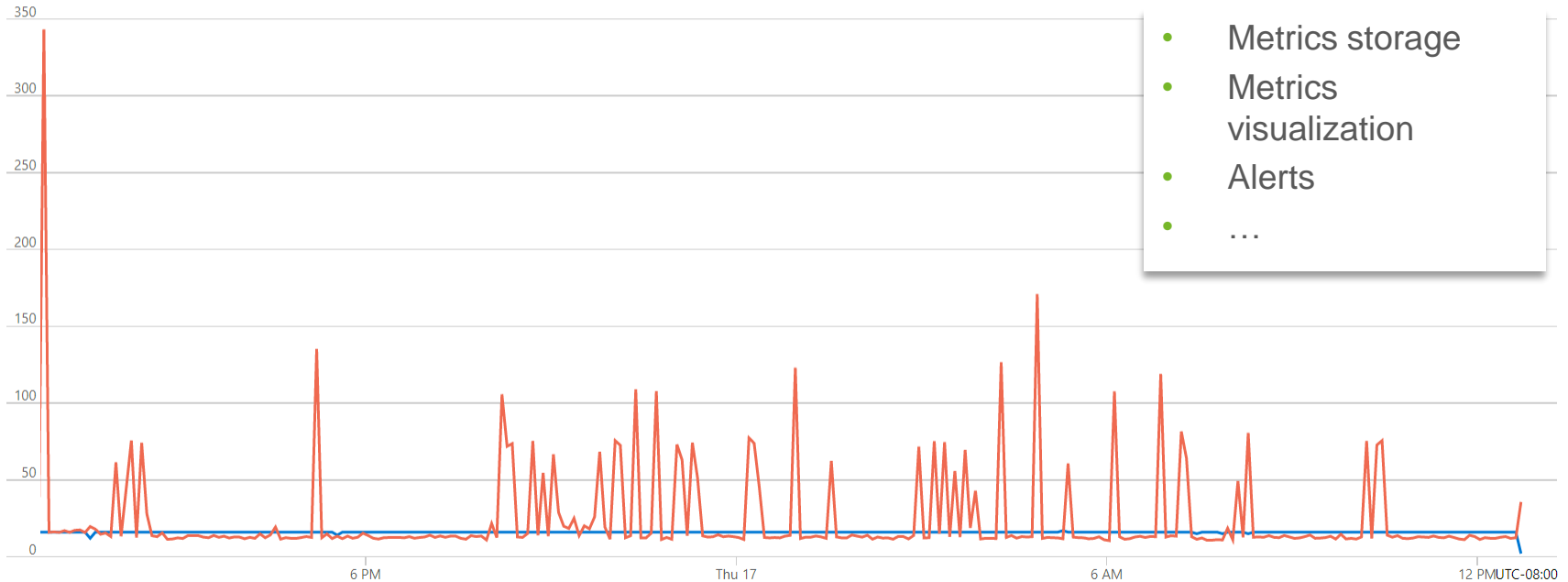
- Metrics storage
- Metrics visualization
- Alerts
- …

Server requests (Sum) tailspintoyswebapp	Server response time… tailspintoyswebapp	Failed requests (Sum) tailspintoyswebapp
4.59 k	**24.30** ms	**0**

Source: https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/data-platform-metrics

RETIT

Auto-Instrumentation Config for AWS and Azure

Add Java agent to Java process, configure service name for telemetry data via env-property:

```
OTEL_RESOURCE_ATTRIBUTES=service.name=MyApp,service.namespace=MyTeam
java -javaagent:path/to/agent.jar -jar myapp.jar
```

Agents

AWS:	https://github.com/aws-observability/aws-otel-java-instrumentation/releases
Azure:	https://github.com/microsoft/ApplicationInsights-Java/releases
OTEL:	https://github.com/open-telemetry/opentelemetry-java-instrumentation/releases

Additional Configurations

AWS:
```
OTEL_TRACES_SAMPLER=parentbased_traceidratio
OTEL_TRACES_SAMPLER_ARG=0.1
OTEL_EXPORTER_OTLP_ENDPOINT=https://mycollectorhost:4317
```

\+ ADOT Collector Setup: https://aws-otel.github.io/docs/getting-started/collector

Azure:
```
APPLICATIONINSIGHTS_CONNECTION_STRING=InstrumentationKey=...
```

Setup GCP OTEL Traces Recording

Dependencies (Gradle):

GCP Trace exporter:
```
implementation 'com.google.cloud.opentelemetry:exporter-trace:0.20.0' (exporter-metrics-0.20.0-alpha)
```

Using only the OTEL SDK:
```
implementation 'io.opentelemetry:opentelemetry-sdk:1.10.1' (opentelemetry-sdk-metrics separate!)
```

```java
import com.google.cloud.opentelemetry.trace.TraceConfiguration;
import com.google.cloud.opentelemetry.trace.TraceExporter;
import io.opentelemetry.sdk.OpenTelemetrySdk;
import io.opentelemetry.sdk.trace.SdkTracerProvider;
import io.opentelemetry.sdk.trace.export.BatchSpanProcessor;

TraceExporter = TraceExporter.createWithConfiguration(TraceConfiguration.builder()
        .setProjectId("MY_GCP_PROJECT_ID").build());
//...
OpenTelemetrySdk opentelemetry = OpenTelemetrySdk.builder()
        .setTracerProvider(SdkTracerProvider.builder()
        .addSpanProcessor(BatchSpanProcessor.builder(traceExporter).build())
        .build()).buildAndRegisterGlobal();
```

Refer to: https://github.com/GoogleCloudPlatform/opentelemetry-operations-java/blob/main/exporters/trace/README.md

Recording Custom OTEL Traces

```java
Tracer tracer =
        openTelemetry.getTracer("instrumentation-lib-name", "1.0.0");

// …

Span span = tracer.spanBuilder("new span").startSpan();
// put the span into the current Context
try (Scope scope = span.makeCurrent()) {
    // … your business code
} catch (Throwable t) {
    span.setStatus(StatusCode.ERROR, "Error Message");
} finally {
    span.end();
}
```

Refer to: https://opentelemetry.io/docs/instrumentation/java/manual/#tracing

Recording Custom OTEL Metrics

```java
// MeterProvider => access required, API in alpha, depends on OTEL release
// Gets or creates a named meter instance
Meter meter = meterProvider.meterBuilder("instrumentation-lib-name")
    .setInstrumentationVersion("1.0.0").build();

// Build counter e.g. LongCounter
LongCounter counter = meter.counterBuilder("my_counter")
    .setDescription("My Counter").setUnit("1").build();

// It is recommended that the API user keep a reference to a Bound
// Counter for the entire time or call unbind when no-longer needed.
BoundLongCounter someWorkCounter = counter.bind(
    Attributes.of(stringKey("Key"), "SomeWork"));

// Record data
someWorkCounter.add(123);
```

Refer to: https://opentelemetry.io/docs/instrumentation/java/manual/#metrics-alpha-only

Cost

Provider (Price Overview Link)	Billed by	Free tier?	cost / unit (/month)	Pricing Calculator
AWS X-Ray, CloudWatch	# traces recorded # traces retrieved # custom metrics, ingested data volume, archived data volume …	Yes	$5.00 / 1 Mio. traces recorded, $0.50 / 1 Mio. traces retrieved, $0.30 / custom metric, $0.63 / 1GB ingested, … (region: EU Frankfurt)	https://calculator.aws/#/createCalculator/xray https://calculator.aws/#/createCalculator/CloudWatch
GCP Cloud Trace, Cloud Monitoring Metrics	# spans recorded, metrics data volume # Monitoring API calls	Yes	$0.20 / Mio. spans recorded, $0.2580 / MiB (first 150-100k MiB, …) …	https://cloud.google.com/products/calculator
Azure Azure Monitor	ingested data volume ("As-You-Go" or Commitment-tiered) # custom metrics, # metrics queries …	Yes	$2.99 / GB (Pay-As-You-Go) $0.258/MB (first 150-100k MB, …) … (region: Germany West Central)	https://azure.microsoft.com/en-us/pricing/calculator/

=> Use the pricing calculators!

Migration to Other Commercial Observability Tools

- OTEL is supported by Dynatrace, AppDynamics, NewRelic, Datadog, …

    - These observability tools can ingest OTEL telemetry.

- Custom OTEL trace/metrics recording requires little to no adaption, due to being vendor neutral.

    - Azure: Exception, since custom recording code is vendor specific.

- ADOT Collector: Out of the box support to configure telemetry export to commercial tools.

Thanks a lot for your attention!

Questions?

lubomski@retit.de