AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

ANT306-R

Beyond monitoring: Observability with operational analytics data

Muhammad Ali (he/him)

Sr. Analytics Specialist Solutions Architect
AWS

Rafael Gumiero (he/him)

Sr. Analytics Specialist SA
AWS

Resolving downtime

Reproducing problems

Performance

Improving customer experience

Creating innovative features

Debugging failures

Scale challenges

Service outages

Networking connectivity

Hardware failure

Availability

Slow response analysis

Cost optimization

Component interdependency

Long mean time to detect failure

Long mean time to repair failure

New releases

Rolling out to new geography

Improve developer experience

Building automations

Monitoring recovery

Resolving insufficient capacity

Reproducing problems

Performance

Improving customer experience

Creating innovative features

Debugging failures

Scale challenges

Service outages

Networking connectivity

Hardware failure

Availability

Slow response analysis

Resolving downtime

Cost optimization

Component interdependency

Long mean time to detect failure

Long mean time to repair failure

New releases

Rolling out to new geography

Improve developer experience

Building automations

Monitoring recovery

Resolving insufficient capacity

Agenda

Observability overview

Observability with Amazon OpenSearch Service

Observability ingestion tools with OpenSearch Service

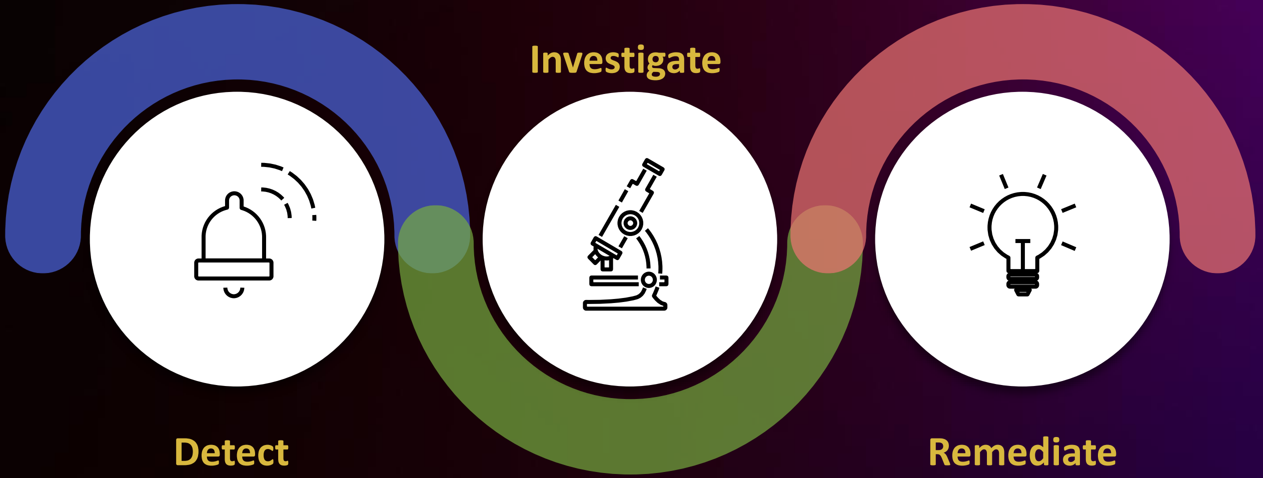
Lab: Microservice Observability with Amazon OpenSearch Service Workshop

Lab setup

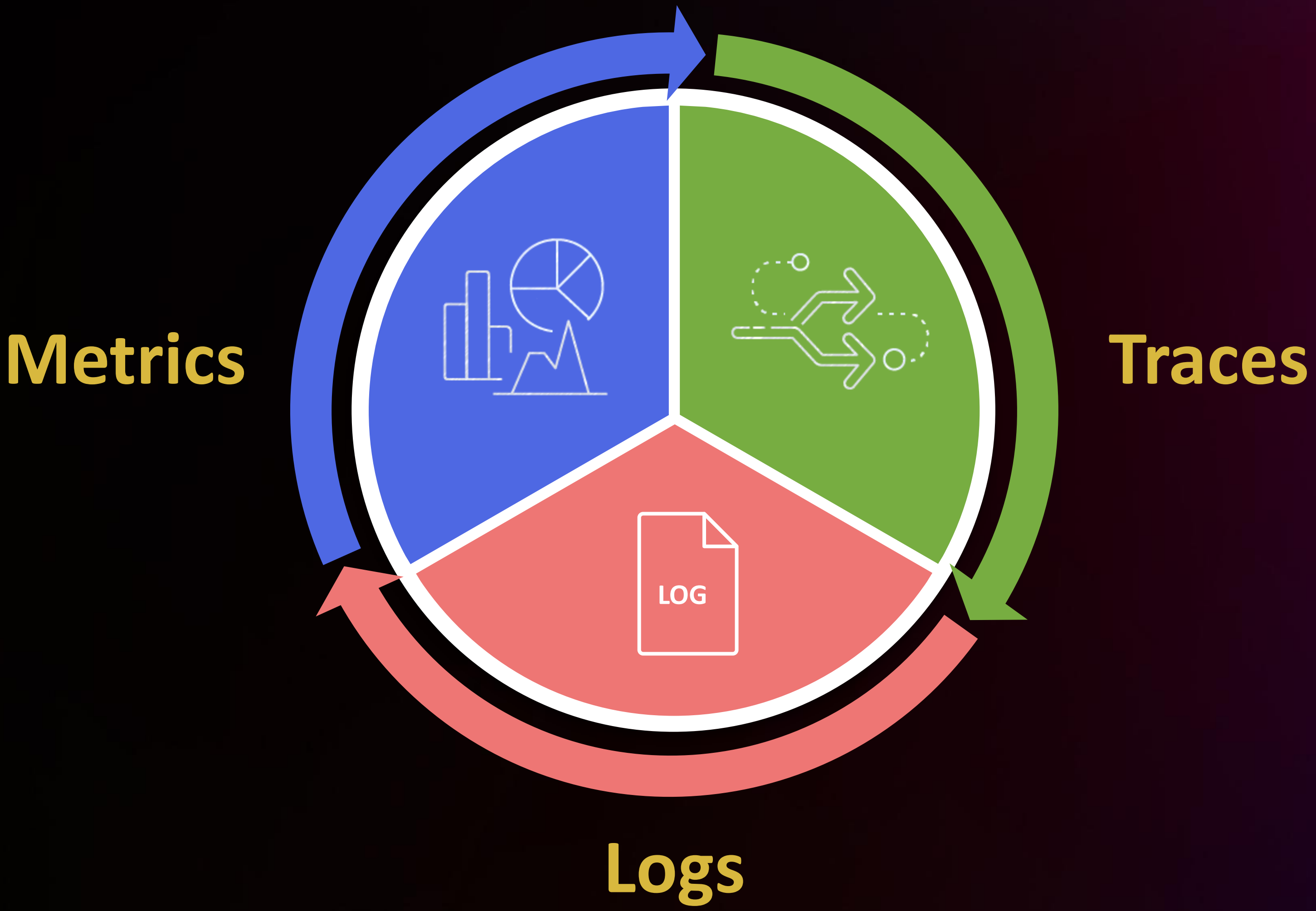
Observability overview

What is observability?

Observability is the ability of a system to be observed by gathering telemetry data to EFFICIENTLY detect, investigate, and remediate issues faster



What are the observability pillars?



What are the main observability use cases?

- Microservices and containers
- Digital experience monitoring (DEM)
- AIOps and DevOps
- Data lakes

Observability with OpenSearch Service

Amazon OpenSearch Service

OpenSearch Service securely unlocks real-time search, monitoring, and analysis of operational data

Managed: Increase operational excellence by using a popular open-source solution

Secure: Audit and secure your data with a data center and network architecture and built-in certifications

Observability: Systematically detect potential threats and react to a system’s state through an open-source solution
for machine learning, alerting,
and visualization

Cost-conscious: Optimize time and resources for strategic work

Observability with OpenSearch Service

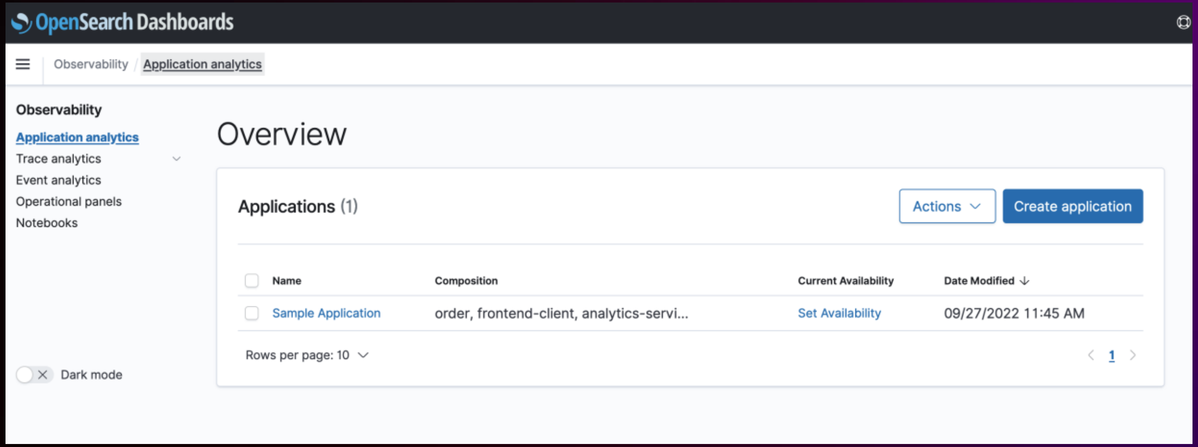
APM – Application analytics to correlate logs, traces, and metrics

Distributed tracing with trace analytics

Log monitoring, event analytics

- Expanded log analytics capabilities
- PPL parse commands support

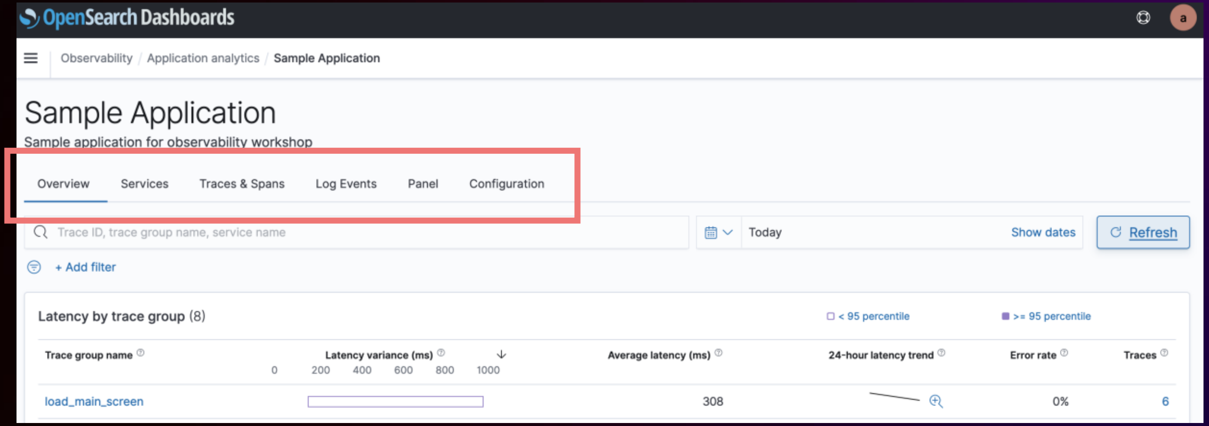
Out-of-the-box operational analytics (coming soon – metric extraction, alerting, anomaly detection, Prometheus integration)



Overview

Applications (1)

Name	Composition	Current Availability	Date Modified
Sample Application	order, frontend-client, analytics-servi...	Set Availability	09/27/2022 11:45 AM



Sample Application

Sample application for observability workshop

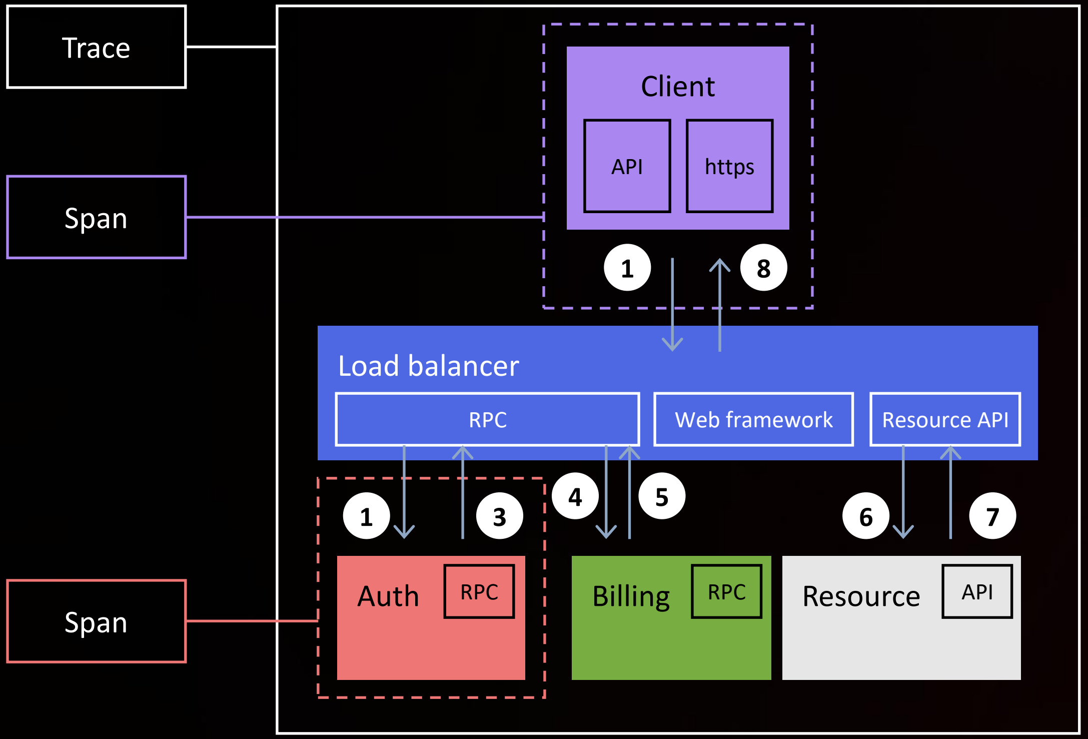
Latency by trace group (8)

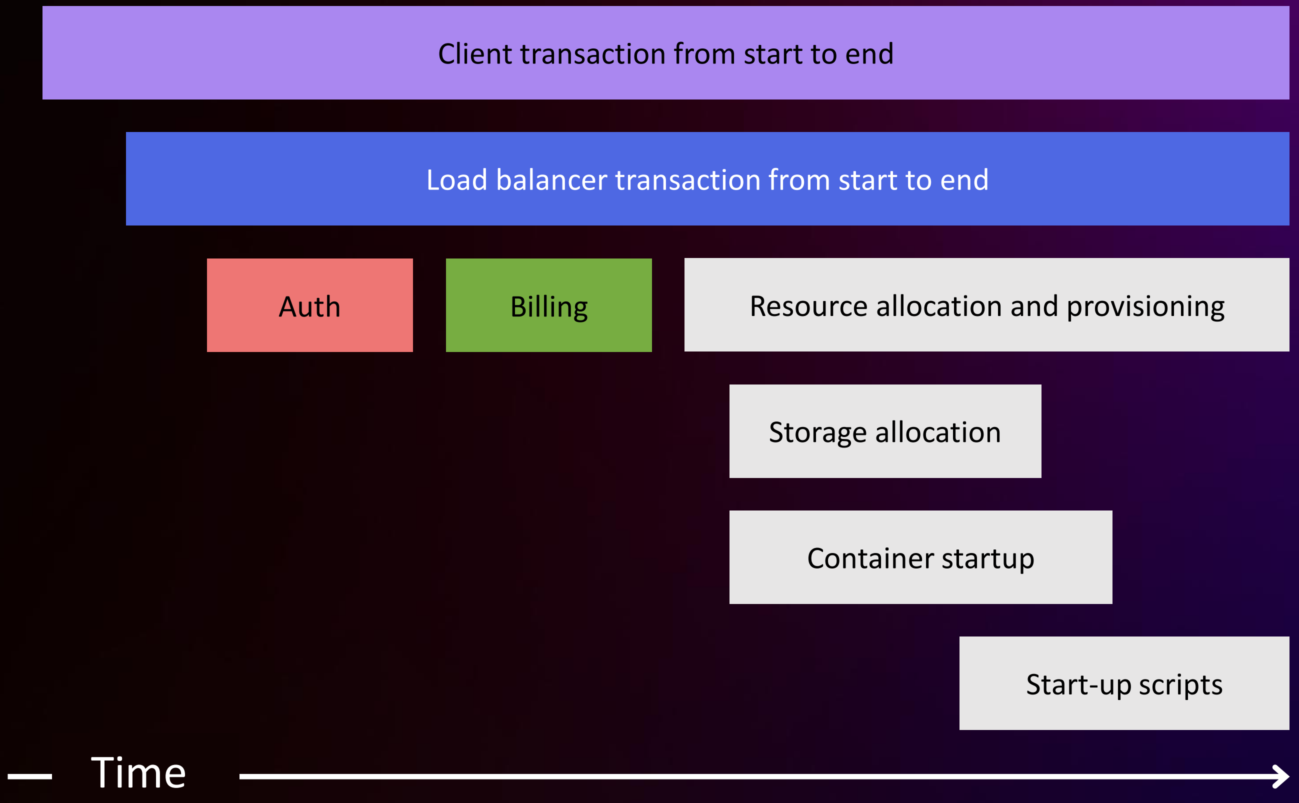
Trace group name	Latency variance (ms)	Average latency (ms)	24-hour latency trend	Error rate	Traces
load_main_screen		308		0%	6

What is distributed tracing?

Distributed tracing refers to methods of observing requests as they propagate through distributed systems

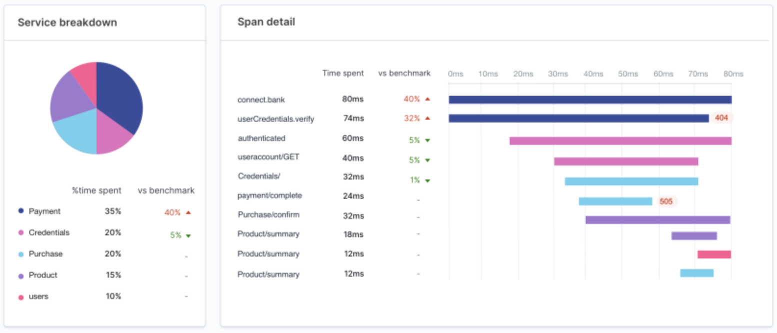
A trace tells the story of a transaction or workflow as it propagates through a system





Client transaction from start to end
Load balancer transaction from start to end
Auth
Billing
Resource allocation and provisioning
Storage allocation
Container startup
Start-up scripts
Time

What is trace analytics?



Service breakdown

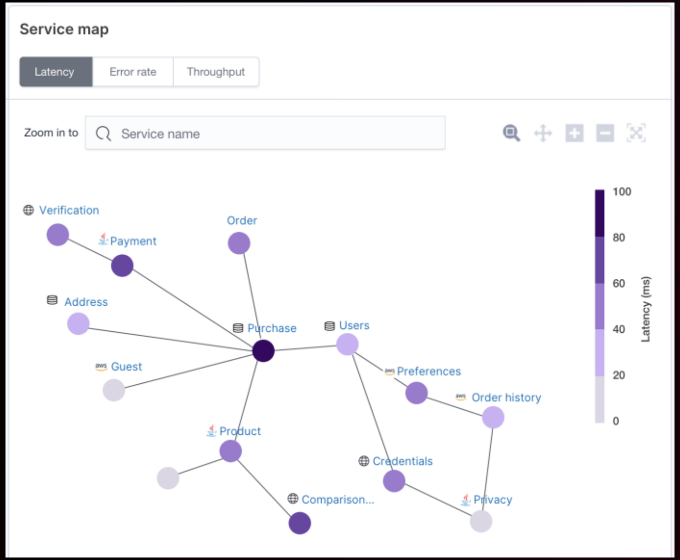
	%time spent	vs benchmark
● Payment	35%	40% ▲
● Credentials	20%	5% ▼
● Purchase	20%	-
● Product	15%	-
● users	10%	-

Span detail

	Time spent	vs benchmark
connect.bank	80ms	40% ▲
userCredentials.verify	74ms	32% ▲
authenticated	60ms	5% ▼
useraccount/GET	40ms	5% ▼
Credentials/	32ms	1% ▼
payment/complete	24ms	-
Purchase/confirm	32ms	-
Product/summary	18ms	-
Product/summary	12ms	-
Product/summary	12ms	-

Trace-span details

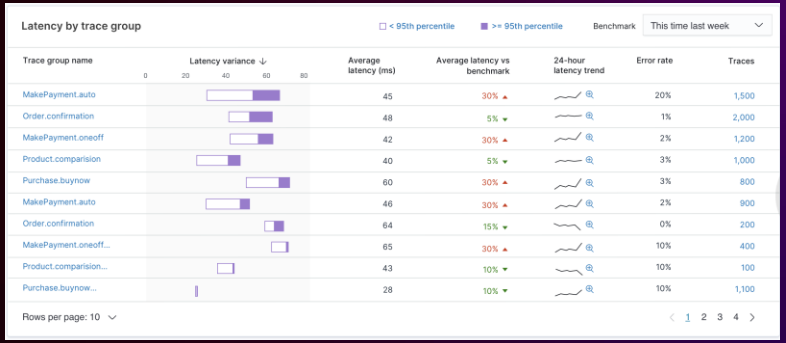
- Single request performance
- Diagnose root cause



Service map

Service maps

- End-to-end view
- Isolate issues to services



Latency by trace group

□ < 95th percentile ■ >= 95th percentile Benchmark This time last week

Trace group name	Latency variance ↓	Average latency (ms)	Average latency vs benchmark	24-hour latency trend	Error rate	Traces
MakePayment.auto		45	30% ▲		20%	1,500
Order.confirmation		48	5% ▼		1%	2,000
MakePayment.oneoff		42	30% ▲		2%	1,200
Product.comparision		40	5% ▼		3%	1,000
Purchase.buynow		60	30% ▲		3%	800
MakePayment.auto		46	30% ▲		2%	900
Order.confirmation		64	15% ▼		0%	200
MakePayment.oneoff...		65	30% ▲		10%	400
Product.comparision...		43	10% ▼		10%	100
Purchase.buynow...		28	10% ▼		10%	1,100

Rows per page: 10

Trace groups

- Monitor performance
- Identify issues early

Observability ingestion tools with OpenSearch Service

Open source instrumentation and transformation

OpenTelemetry

- OpenTelemetry is a community-driven, open-source project designed for the creation and management of **telemetry data** such as traces, metrics, and logs
- Supports many popular open source wire formats including Jaeger, Zipkin, and Prometheus
- Currently supports traces (GA), metrics (GA), and logs (experimental)

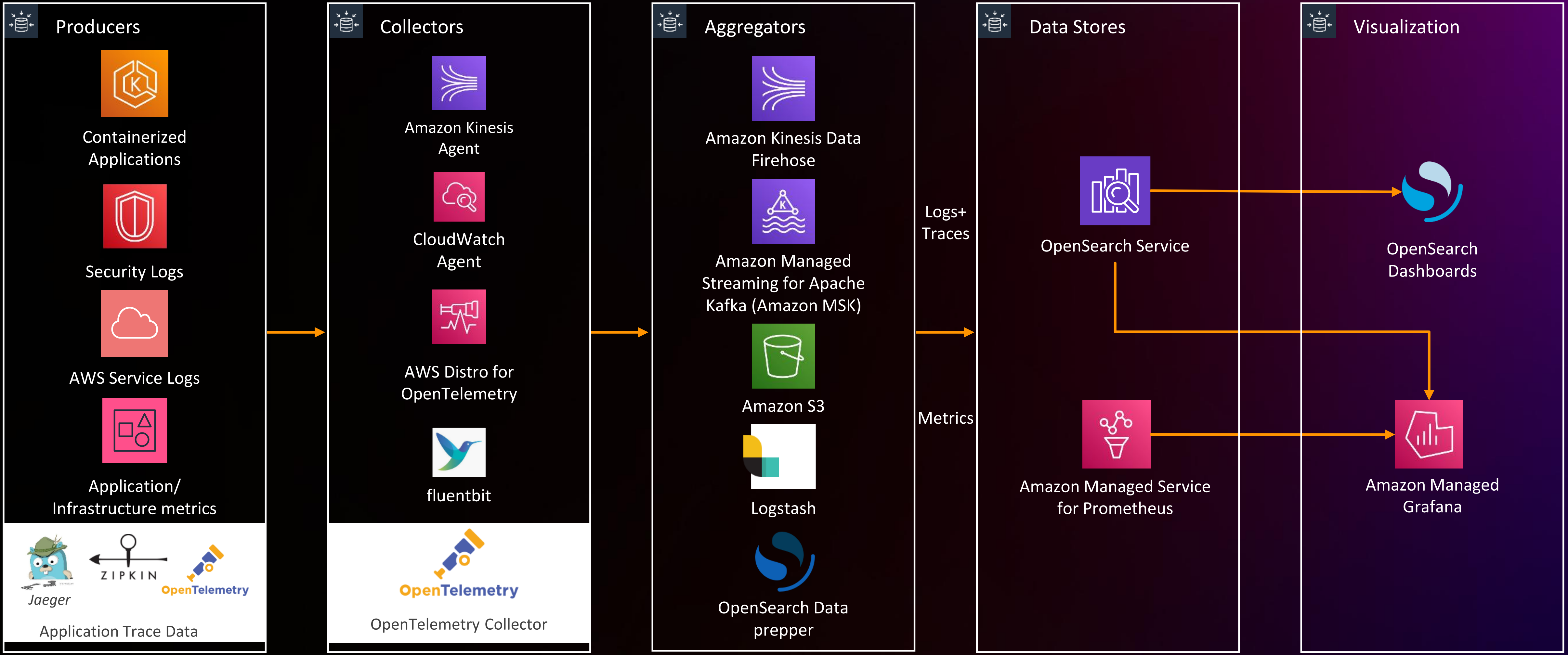
fluentbit

- Fluent Bit is a community-driven, open source, super fast, lightweight, and highly scalable logging and metrics processor and forwarder
- Plugin-based ecosystem that collects, filters, transforms and augments data for logs, metrics, and traces
- Extensible and integrates with Prometheus, OpenSearch Service, Amazon CloudWatch, AWS X-Ray, Amazon S3, Amazon Kinesis Data Streams, and other AWS managed open-source offerings

Data Prepper

- Data Prepper is a community-driven, open source data collector for processing observability data
- Provides features to filter, enrich, transform, normalize, and aggregate data for downstream analytics and visualization
- Currently supports processing of distributed trace data and log ingestion with plans to support metric data in the future
- Integrations with Jeager, Zipkin, OpenTelemetry, and Fluent Bit

Observability ingestion flow

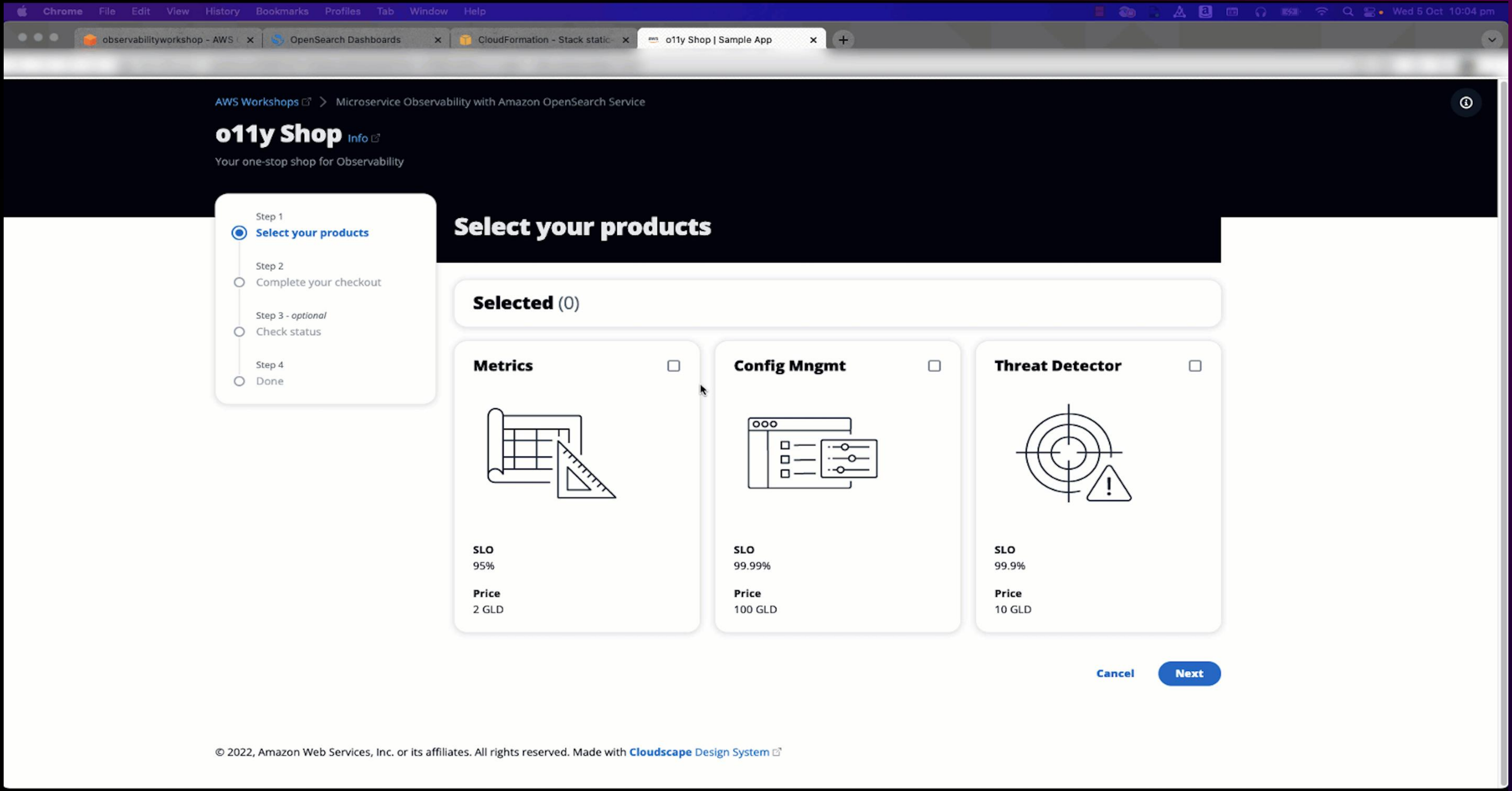


Lab: Microservice observability with Amazon OpenSearch Service workshop

Workshop objectives

1. Discover new observability capabilities in OpenSearch Service
   - Trace analytics and application analytics
   - Log-trace correlation (metrics coming soon)
   - Interactive log visualization
2. Use data collection and ingestion technologies
   - AWS Distro for OpenTelemetry to collect traces
   - Fluent Bit to collect logs
   - OpenSearch Data Prepper for transformation and ingestion
3. Perform a root cause analysis using application analytics

Use case statement


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Wed 5 Oct 10:04 pm
observabilityworkshop - AWS
OpenSearch Dashboards
CloudFormation - Stack static-
o11y Shop | Sample App

AWS Workshops > Microservice Observability with Amazon OpenSearch Service

o11y Shop Info

Your one-stop shop for Observability

Step 1
● Select your products

Step 2
○ Complete your checkout

Step 3 - *optional*
○ Check status

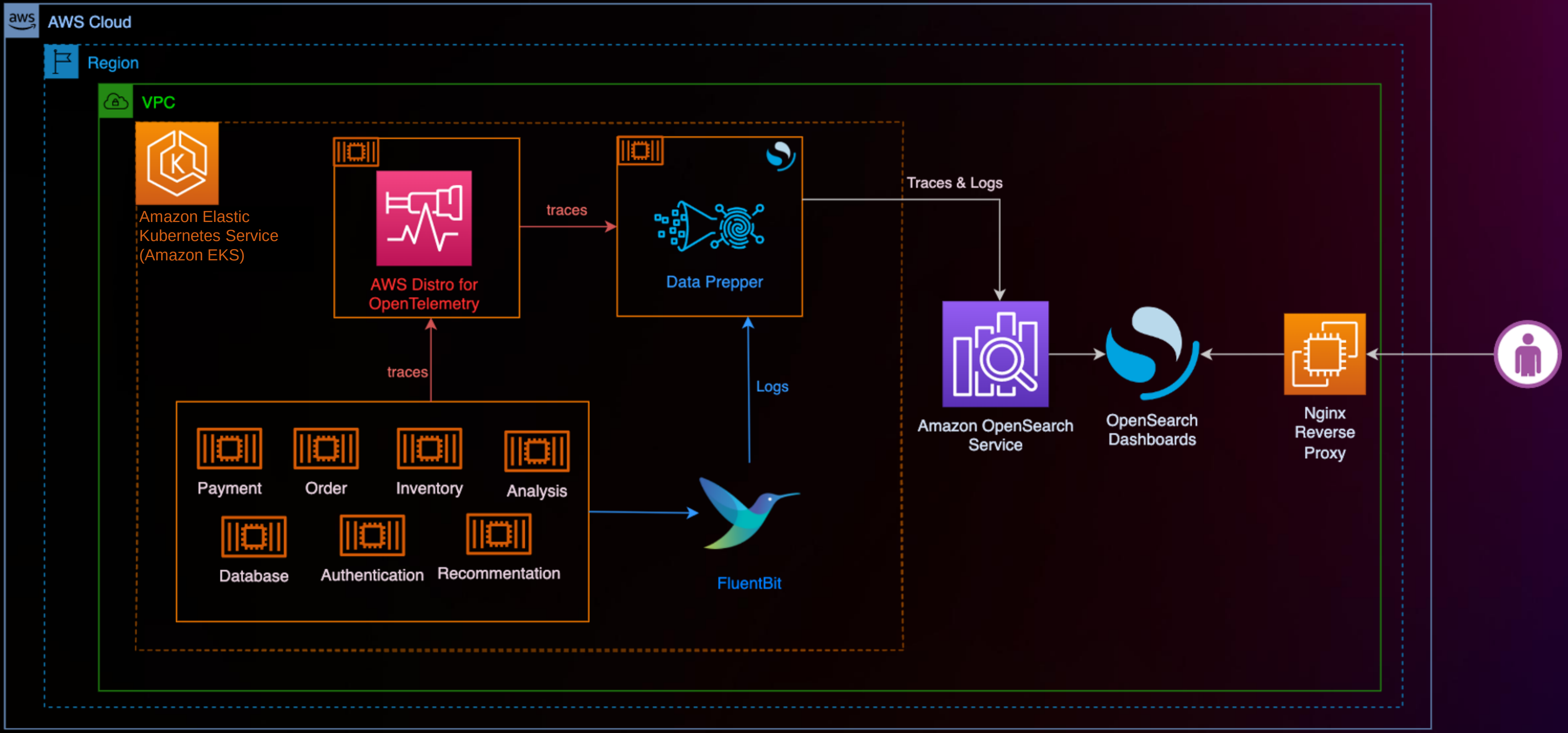
Step 4
○ Done

Select your products

Selected (0)

Metrics □	Config Mngmt □	Threat Detector □
SLO 95%	**SLO** 99.99%	**SLO** 99.9%
Price 2 GLD	**Price** 100 GLD	**Price** 10 GLD

Workshop architecture



Root cause analysis workflow

1. Define your observability application composition: log sources, services, and entities
2. Observe the latency, error rate, and trends associated with each trace group and service
3. Filter the traces and spans involved in the failing service
4. Correlate the log events and the trace data to identify the issue
5. Fix the code line identified in error log message
6. Observe the system after changes

Lab setup

How to access the workshop lab

Lab instructions

https://bit.ly/3kDk9vp

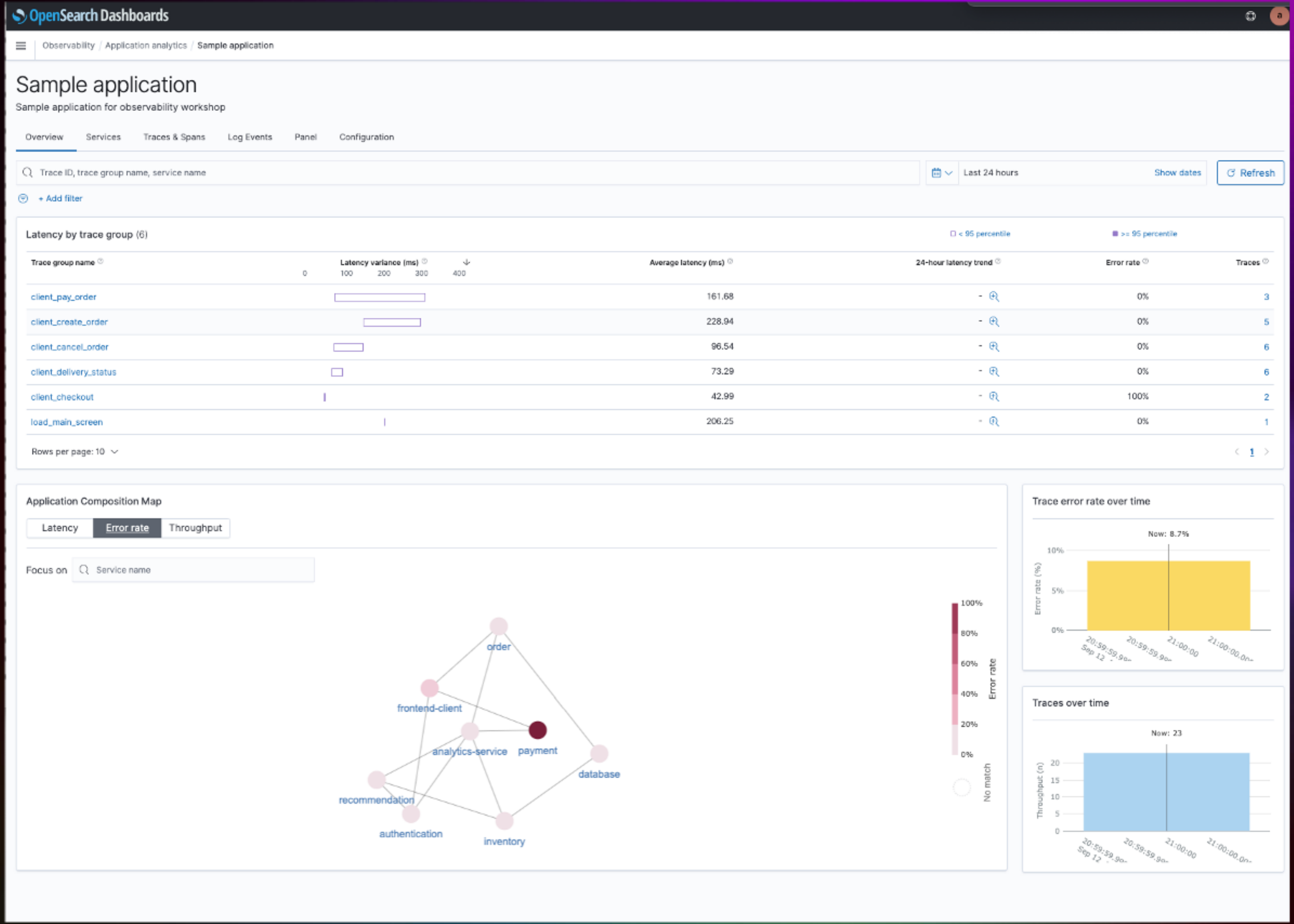
Lab access

http://bit.ly/3giC9Nb

Access Code: 2596-03af0c-97

Let’s build!

Checkpoint



Sample application

Sample application for observability workshop

Latency by trace group (6)

□ < 95 percentile ■ >= 95 percentile

Trace group name	Latency variance (ms) 0 100 200 300 400	Average latency (ms)	24-hour latency trend	Error rate	Traces
client_pay_order		161.68	-	0%	3
client_create_order		228.94	-	0%	5
client_cancel_order		96.54	-	0%	6
client_delivery_status		73.29	-	0%	6
client_checkout		42.99	-	100%	2
load_main_screen		206.25	-	0%	1

Rows per page: 10

Thank you!

Please complete the session survey in the **mobile app**