ORACLE

Observability for Cloud Native Apps

Benefits of Cloud Native Applications

Why Cloud Native Apps?	Benefit
Greater Agility	Accelerate the delivery of new high-quality services using CI/CD and automation
Faster time to market	Build, iterate, and deploy apps faster
Scalability	Scale out and in automatically
Platform and language Agnostic	Open Source and standards based, portable Choice of programming languages and frameworks to best address the app needs
Security	Best practice security by default Easy for anyone to run apps at a high security posture

Ineffective Monitoring Can Put Your Modernization Projects at Risk

Do these challenges sound familiar?	*The opportunity cost*
Heterogeneous and multi-language stack	Hard to find expertise for mixed technologies
Data silos complicate analysis	Problem detection and resolution is difficult
Point-solutions lack full-stack visibility	Multiple tools needed for troubleshooting
Mix of on-premises and cloud deployments	Separate on-premises and cloud operations
Lack of interoperability between tools	High integration costs

Observability

Cloud Native Apps Require Re-tooling

Existing monitoring tools are fragmented

- Focus on individual components
- Language and technology specific
- Aggregation and sampling at the source

Microservices architecture drives a focus shift toward cross component monitoring

- Lightweight agents and built-in instrumentation
- Manage together, not individually
- Open standards, multi programing languages
- CI/CD friendly agent and configuration

The need for total Observability

- Trace all transactions, monitor and log everything
- AI/ML based analysis

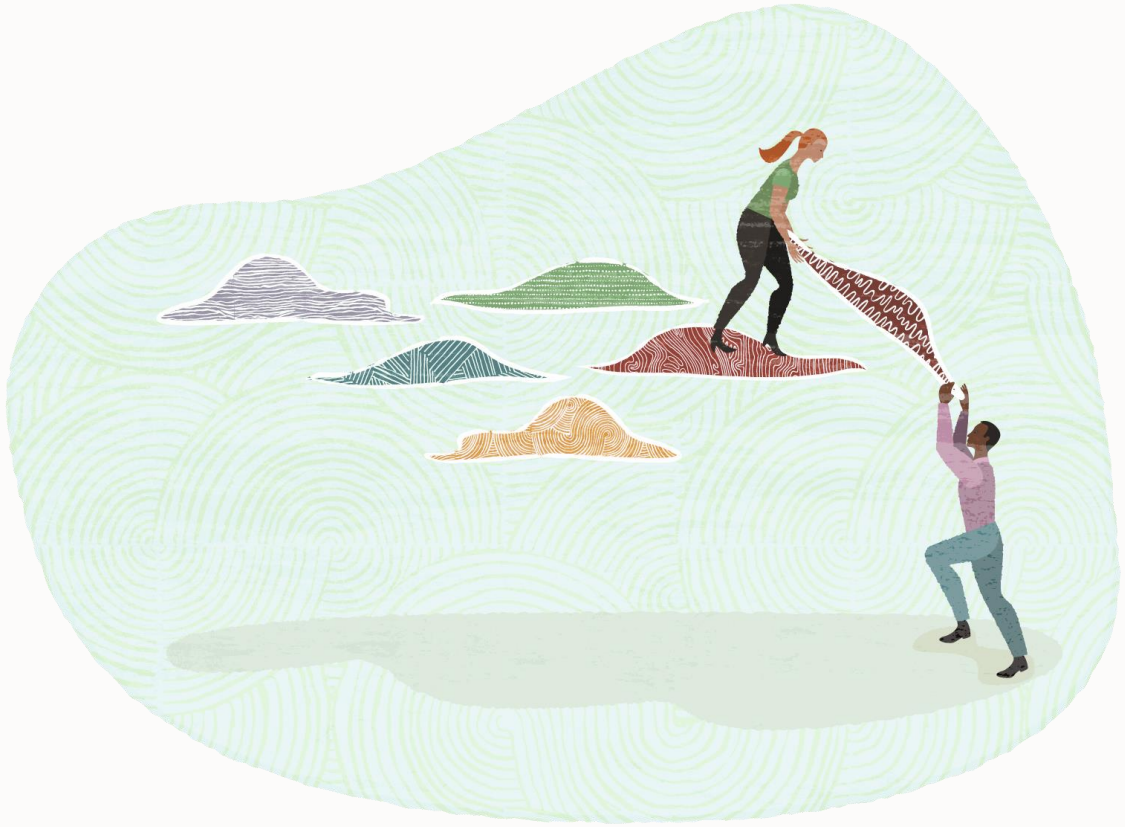
Key Use Cases to Boost Your Observability and Management

Application debugging and distributed profiling
- Solution: Application Performance Monitoring

Digital Experience Management
- Solution: Application Performance Monitoring

Infrastructure and application log analytics
- Solution: Logging Analytics

Performance analytics
- Solution: Operations Insights



Application debugging and distributed profiling

Application Performance Monitoring (APM): Overview

Distributed tracing

- Full capture and analysis of all traces for fine-grained problem diagnostics
- In-context drill-downs from browser events across tiers

Server monitoring

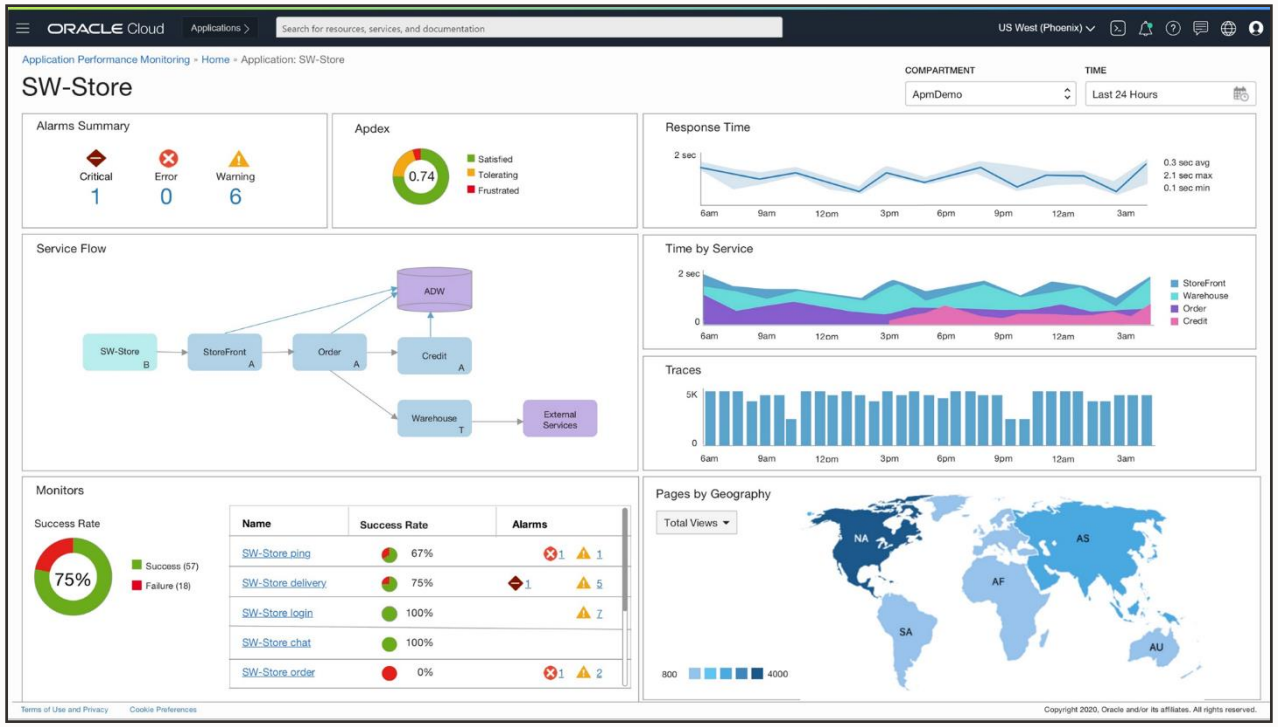
- Performance, availability and load analysis with rich set of server, and transaction metrics

Customizable Dashboards

- Create your own visualizations including data from other Observability & Monitoring Services

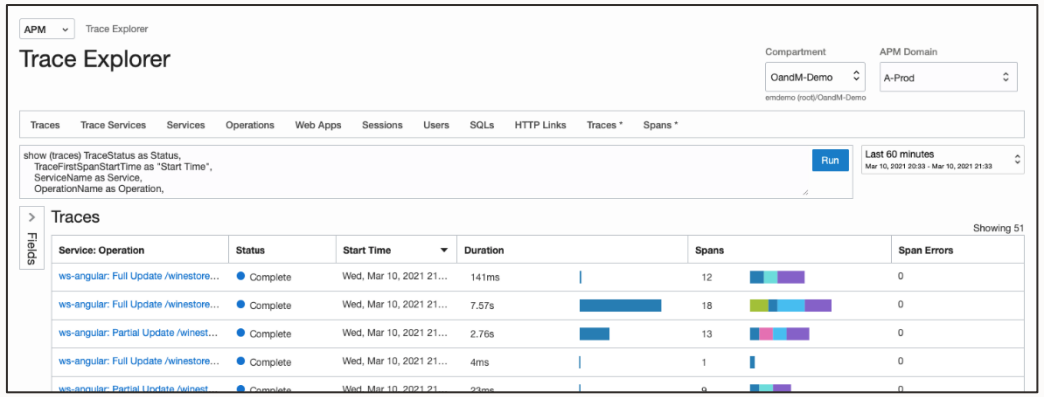
Alerts and Notifications

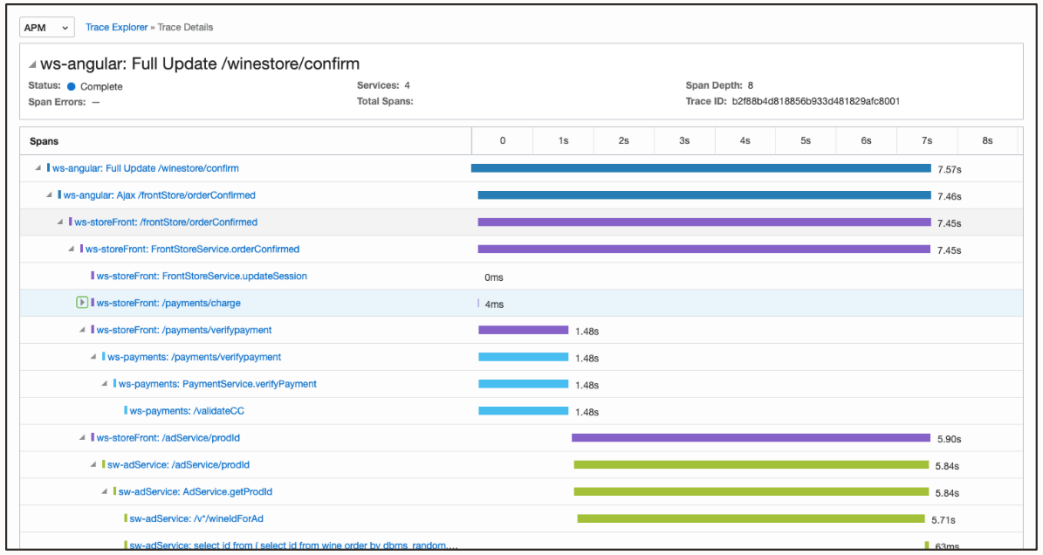
- Utilize the OCI monitoring, notification, service hub services



End-to-End Transaction Tracing and Service Diagnostics

- Capture all steps (spans) of all transactions (traces) all the time
- Trace sync and async transactions, from browser to database
- Accept and store 100% of the traces
- Long term data retention for diagnostics, comparison, and analytics
- Explore and analyze trace data using an intuitive, strong query-language based UI
- Out-of-the-box and customizable tagging (dimension) for rich, meaningful, segmentation
- Service topology discovery and visualization
- Dedicated, customizable views for different consumers (SRE, ops, dev, business, etc.)
- Alerts, notification and automation





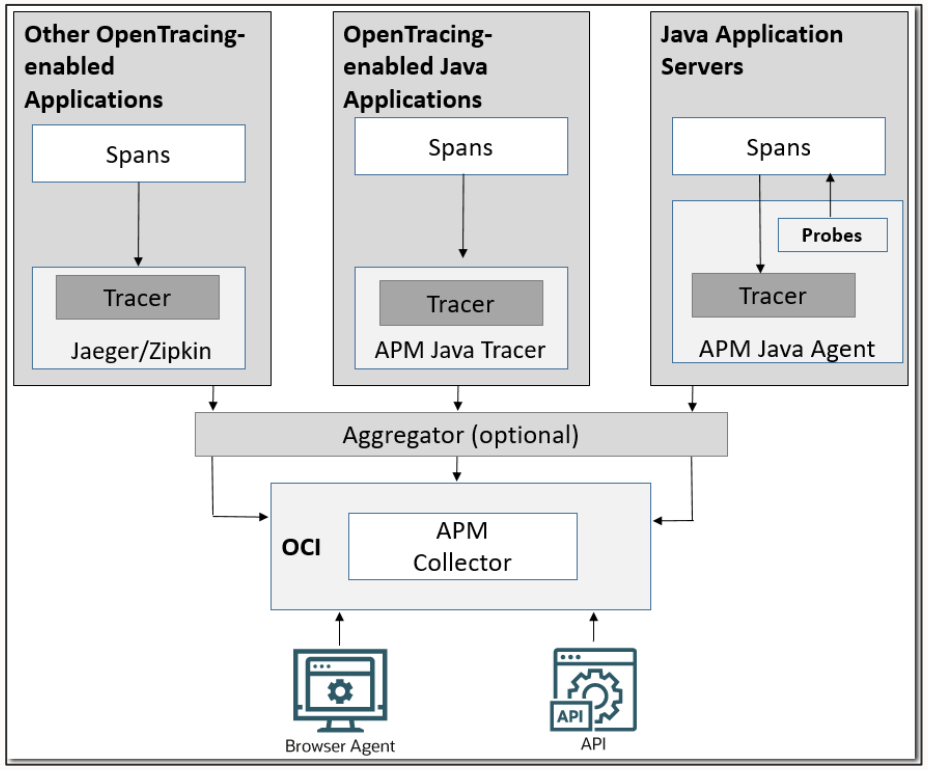
End-to-End Coverage – monitor Browsers, Microservices and Functions

Distributed Tracing

- OpenTracing/OpenTelemetry support
- Automatic instrumentation for Java
- Automatic instrumentation on the browser
- Oracle Cloud Function tracing
- CI/CD integration

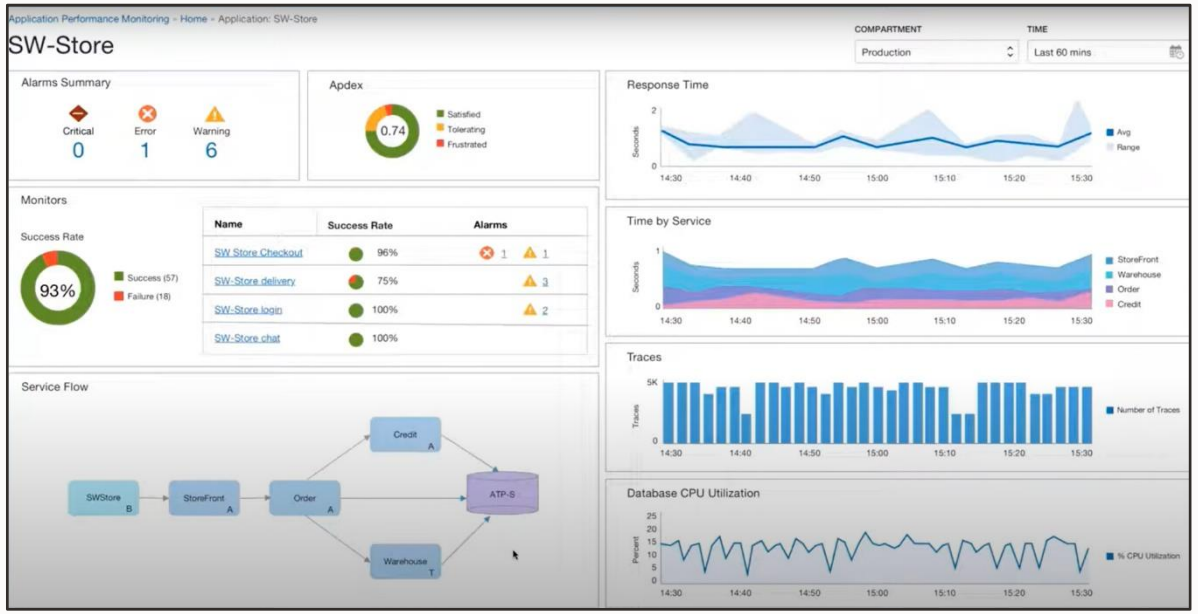
Metrics

- JVM and AppServer metrics
- Prometheus like metric collector
- Out-of-the-box and customizable metric calculated on in-stream span data
- Apdex value for any operation



Demo: Quickly diagnose performance problems

APM provides synthetic monitoring in addition to real user monitoring to deliver a comprehensive application performance monitoring solution. With synthetic monitoring, IT operations teams can proactively monitor their application's availability and detect for early detection of issues regardless of user experience or activity.



Demo: Quickly Diagnose Performance problems

Digital Experience Management

Application Performance Monitoring: Overview

Real User Monitoring

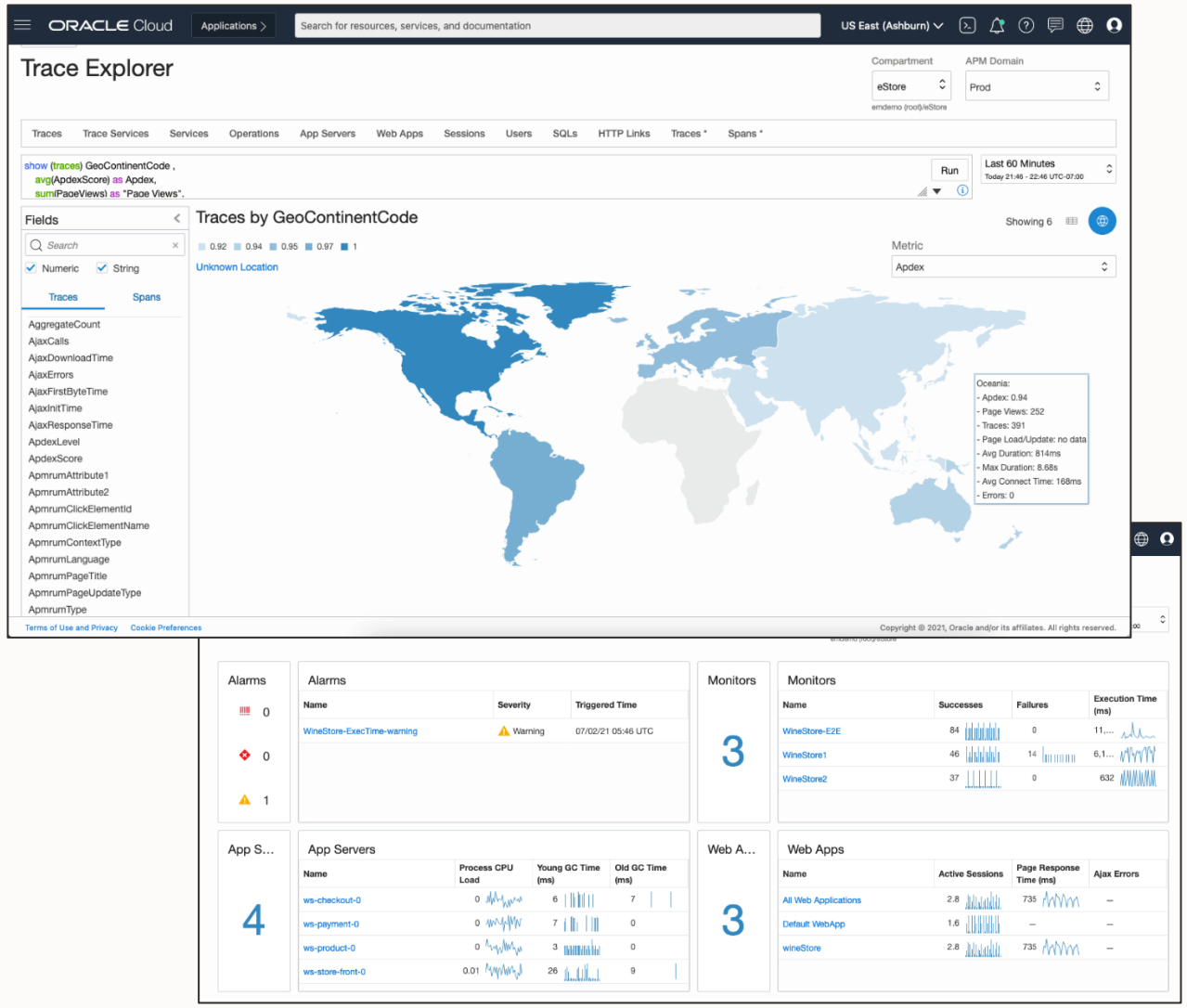
- Measures end-user performance from browser to application
- Deep monitoring with in-context drill-downs from browser events across tiers

Synthetic Monitoring

- Runs pre-defined monitors for proactive monitoring
- Use global and/or dedicated vantage points

Customizable Dashboards

Alerts and Notifications



Real User Monitoring

Browser agent via JavaScript
- Manually inserted to the application

Generate and send spans directly to the APM collector for each:
- Page load
- Page updates
- Ajax calls

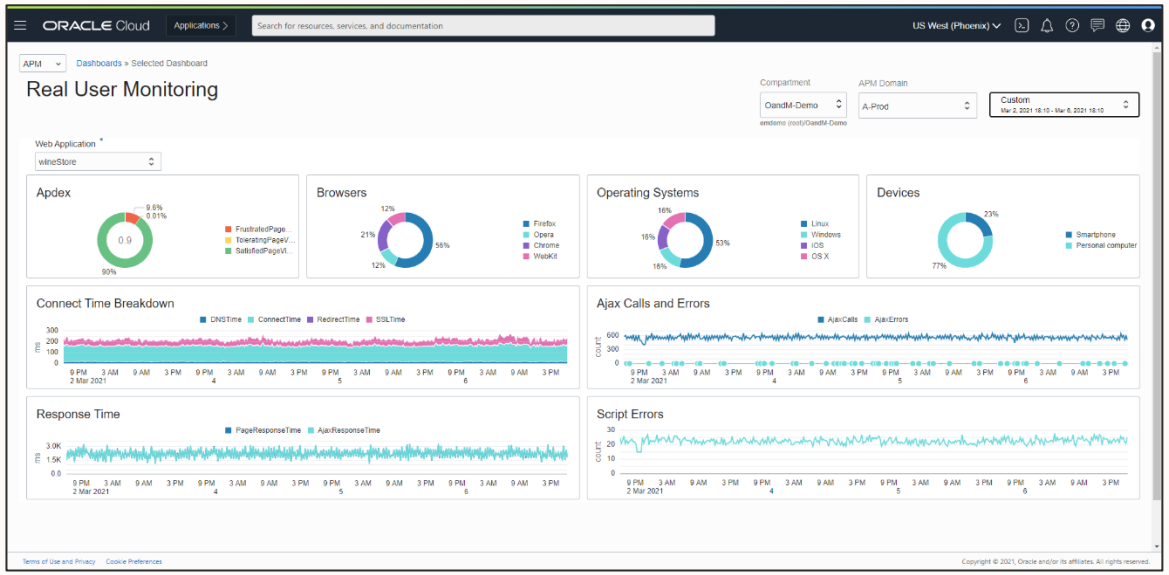
Metrics per span:
- Load time, time to first byte, errors, size, etc.

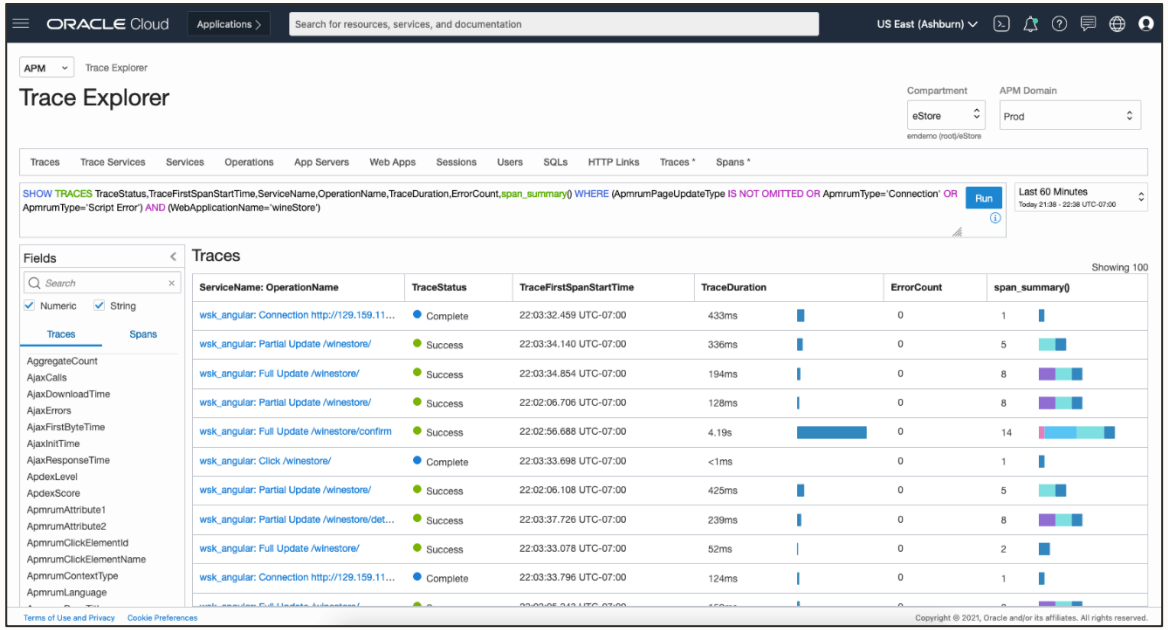
Session Diagnostics (single user session reporting)

Include dimensions collected from the browser
- Device Type
- Browser type and version
- IP Address (location/ISP)

Metrics are available in the monitoring service UI
- Visualize, set and manage alerts

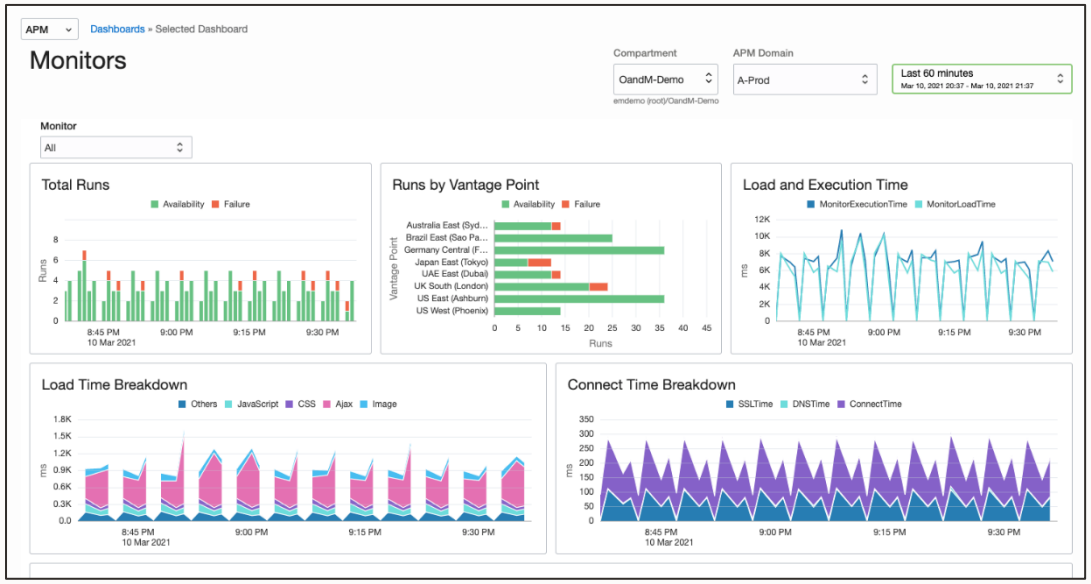
Metrics are available in RUM dashboard

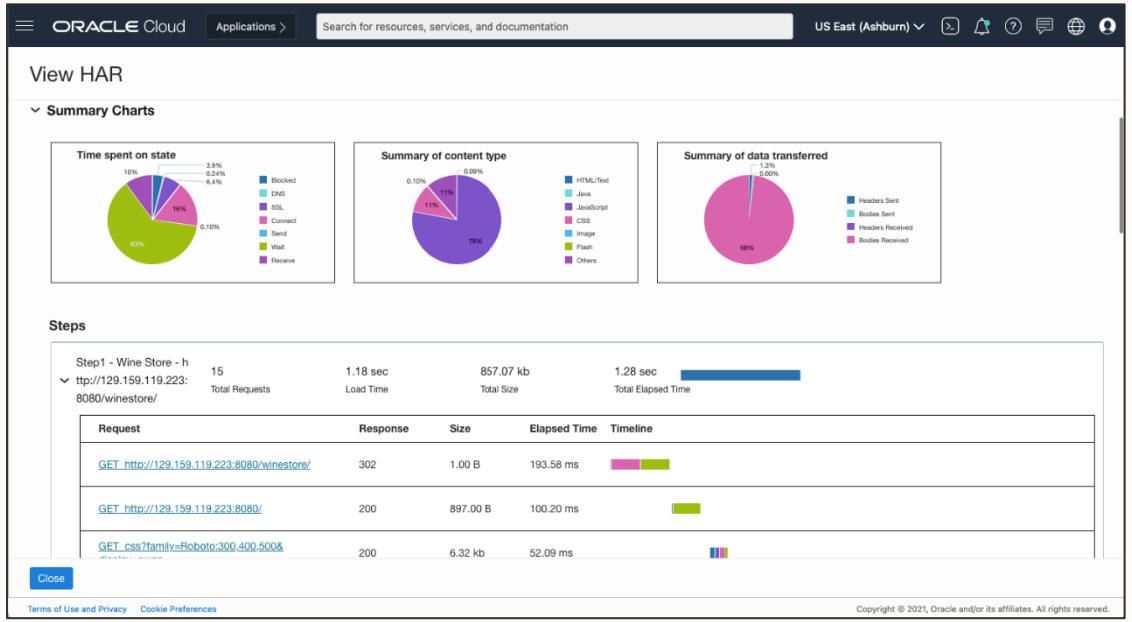




Synthetic Monitoring

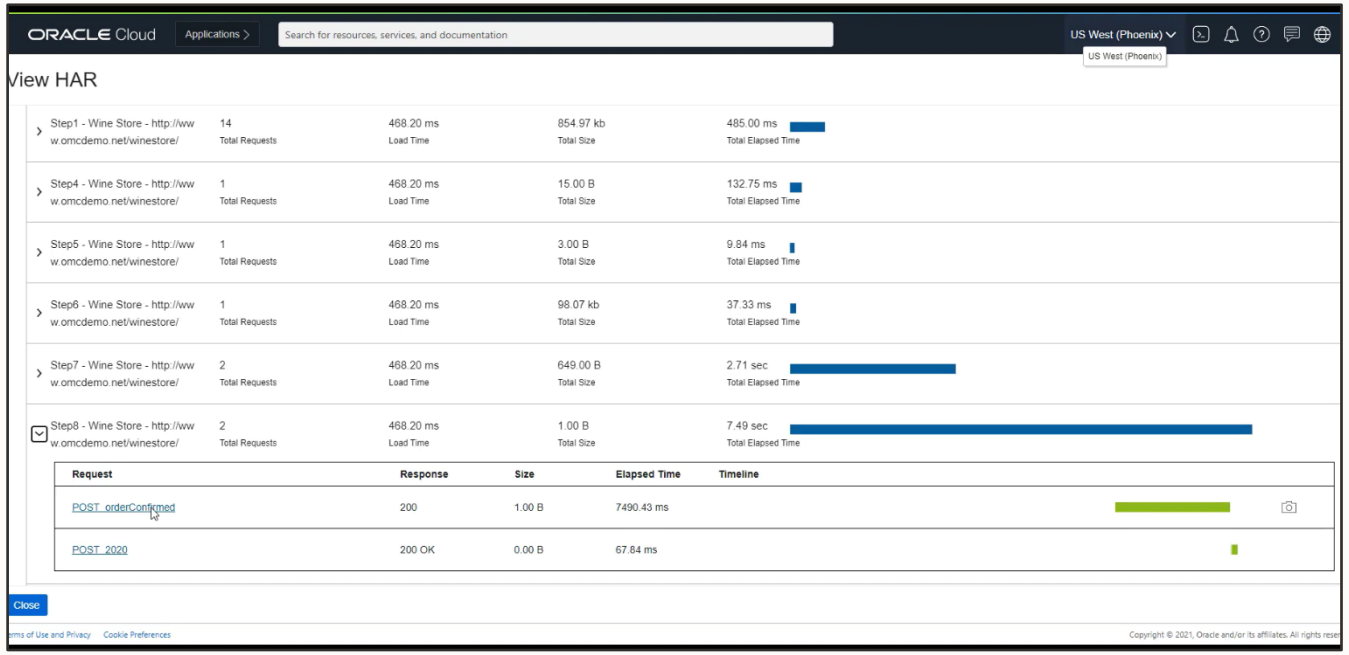
- Capability to run scheduled Monitors
  - Scripted Browser Monitor (Selenium test scripts)
  - Browser Monitor
  - Scripted REST Monitor (Postman scripts)
  - REST Monitor
- Collect and retrieve HAR file & Screenshots
- Oracle hosted Vantage Points (to execute Monitors)
- Combined with server-side tracing
  - The actions of each monitor run are connected to the back-end trace, span collection
- Metric collected for each Monitor run:
  - Load time, time to first byte, errors, size, etc.
- Metrics are available in the monitoring service UI
  - Visualize, Set and manage alerts
- Metrics are available in Monitor dashboard





Demo: Proactive application availability monitoring

APM provides synthetic monitoring in addition to real user monitoring to deliver a comprehensive application performance monitoring solution. With synthetic monitoring, IT operations teams can proactively monitor their application's availability and detect for early detection of issues regardless of user experience or activity.



Demo: Synthetic monitors integrated with Distributed Tracing(5:11)

Log Analytics: Infrastructure & Apps Monitoring

OCI Logging Analytics: Overview

Out-of-Box Knowledge Contents

- Rich collection (250+) of parsers and knowledge enrichment for Oracle and non-Oracle stacks
- Error Categorization

Intuitive data visualization and organization capabilities

- Analyze and visualize based on entity relationship
- In context drill down for troubleshooting

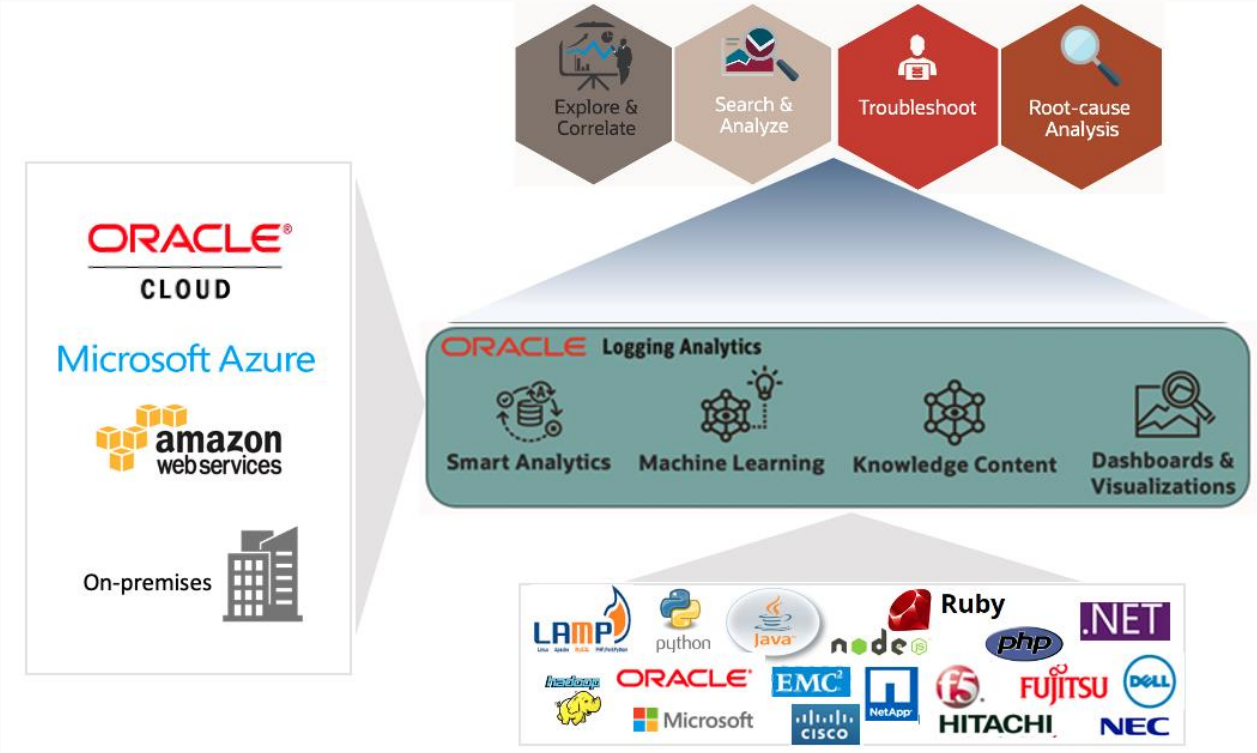
Machine learning based Smart Analytics

- Cluster and Time Series Analysis
- Comparison and Outlier Detection
- Transaction Sequencing, Aggregation and Rollup, Potential issues
- Multi-Dimensional Data Exploration

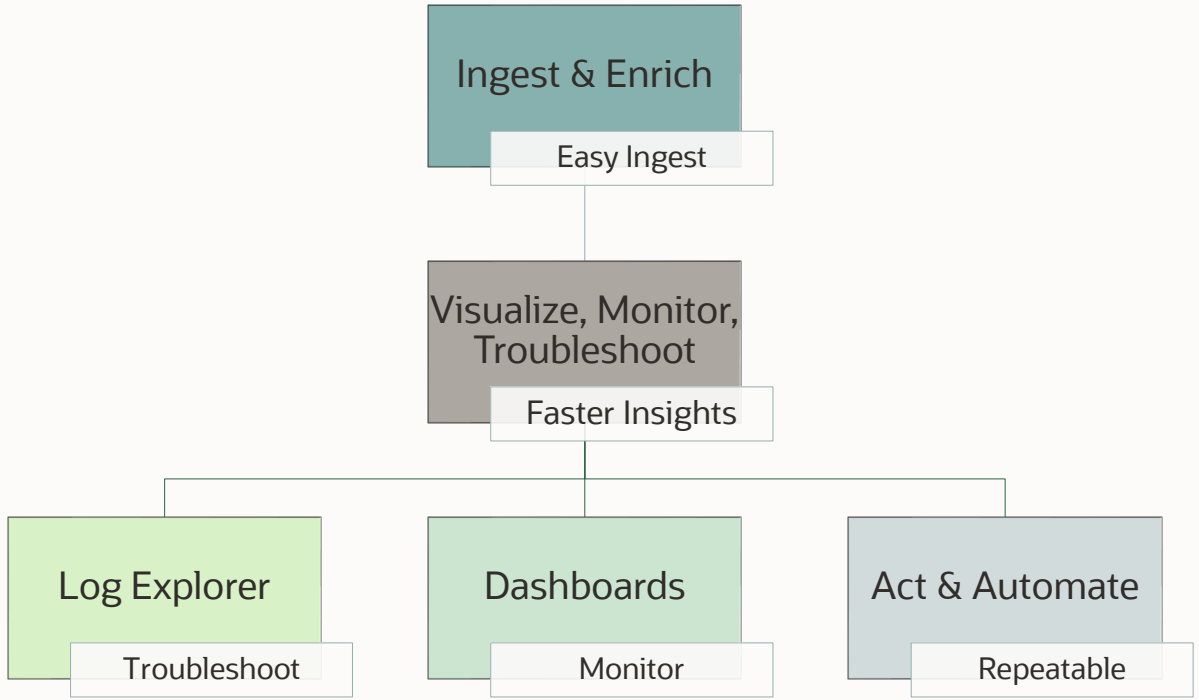
Dashboards for monitoring and reporting

- Create custom dashboard using user created widgets for Logs and metrics data

Ingest, Analyze and Troubleshoot using Log Data



Get Faster Resolution and Value with Logging Analytics



Intuitive & advance visual analytics using curated machine learning algorithms

Pre-defined dashboards & searches to start monitoring in minutes

Pre-defined alerts & integration with other applications & services

One-Click Ingestion Flows enabled by deep application & infrastructure knowledge

Start visualizing, monitoring or troubleshooting with pre-configured analytics knowledge content & dashboards

Native OKE Infrastructure & Apps Log Monitoring

Topology based analysis

Business Service View

- Fulfillment Service

IT Service View

- Order, payment, inventory, shipment apps

Infrastructure View

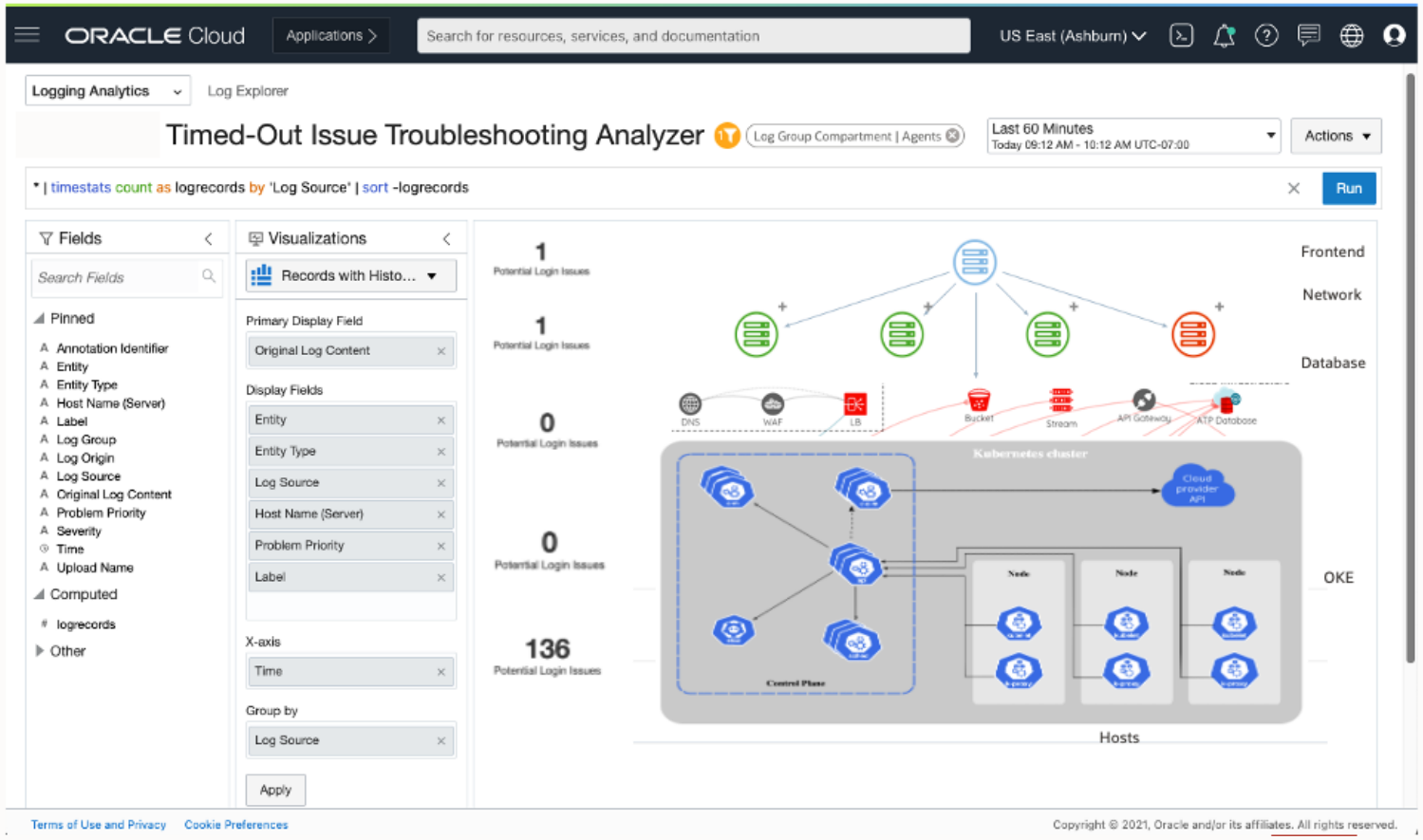
- Nodes, OKE API Server, OKE control plane logs, load balancers, database, pod network map

Automatic entity creation, source identification

Auto-scale monitoring as your OKE scales

Pre-built dashboards, searches at different levels to kick-start analysis

ML based analytics for faster root cause analysis, correlation, and troubleshooting



Performance Analytics

Oracle Cloud Infrastructure Operations Insights Service: Overview

Oracle SQL Warehouse

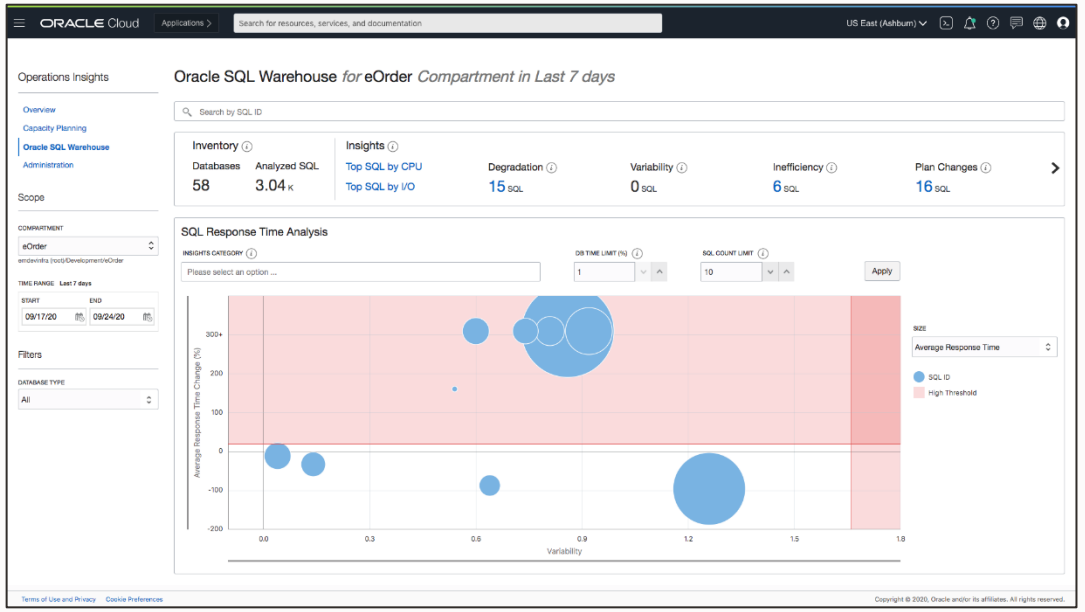
Long term SQL store for Oracle Autonomous Databases and external on-premises database fleet

- Enabling application SQL performance analysis up to 25 months historical data

Find common patterns of SQL performance issues across fleet of Oracle Autonomous Databases and external on-premises databases

Automated insights into SQL performance

- Top resource consuming SQL
- Degraded and unpredictable performance
- Application inefficiency and plan volatility

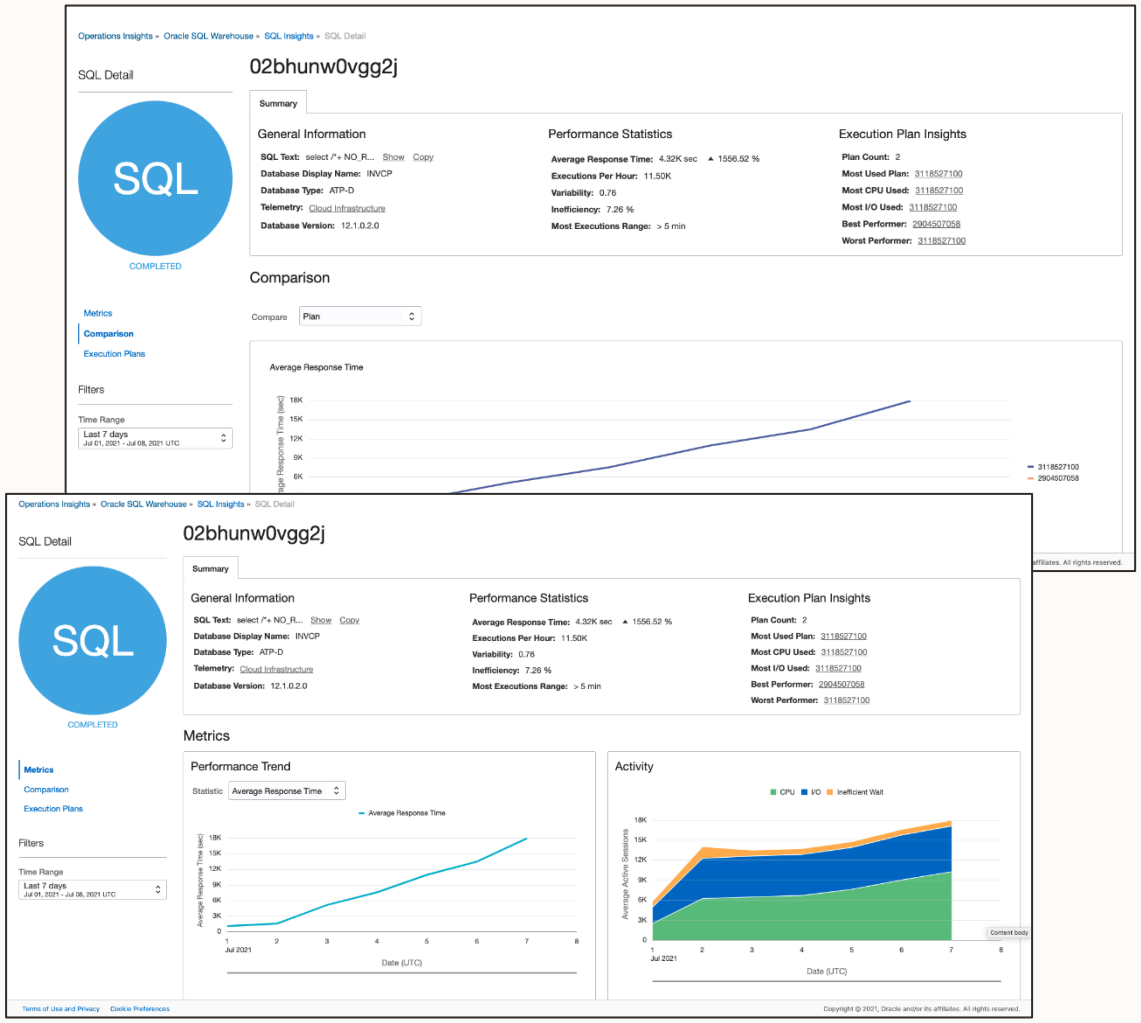


Oracle Cloud Infrastructure Operations Insights Service: Key Capabilities

SQL Performance Insights

Performance details for the selected SQL

- Insights into performance trend, activity, response time
- Compare SQL performance by execution plan hash value or databases based on Average Response Time, Average Active Sessions, Executions Per Hour, I/O Time, and CPU Time
- Compare SQL execution plans
- Quickly identify degraded and unpredictable performance
- Analyze application inefficiency and plan volatility of the SQL



Partners on Oracle Cloud Observability and Management

"Our customer-centric business relies on Oracle Cloud Infrastructure Application Performance Monitoring to deliver exceptional user experience by using a single pane of glass to trace application performance and drill down to root cause of the problem 20% faster."

Roberto Zona

Managing Partner

"Adopting DevOps just got easier. Oracle Cloud Infrastructure's one-click integration with PagerDuty allows customers to leverage their existing investments in incident management."

Steve Gross

Senior Director, Strategic Ecosystem

"SMS messaging from Twilio provides connectivity to over 180 countries. Using this capability on OCI, customers can reach their global, distributed workforce easily while keeping costs low."

Ott Kaukver

CTO

"We're excited to announce two Grafana certified plugins for Oracle, making it easier for our customers to gain end-to-end visibility of any application, database, and infrastructure on OCI—in addition to other environments."

Anthony Woods

CTO

Customers on Oracle Cloud Observability and Management

promata

"We are using machine learning to cut troubleshooting times and the risk of application downtime by 50%."

Michael Wolf
Managing Director

Lone Star College

"We improved our student's experience by reducing performance glitches by 90%, and lowered our effort and costs."

Longin Gogu
Associate Vice Chancellor, Enterprise Applications

drop tank

"As we move more applications to Oracle Cloud, we are excited to use Oracle's Observability and Management solution to eliminate manual effort."

Timothy Miller
CTO

KINGOLD 僑鑫

"Our developers are able to identify code level performance issues 10 times faster."

Steven Chang
CIO

Why customers are choosing Observability & Management

1. Natively built within Oracle Cloud Infrastructure
2. All the services DevOps need to monitor cloud native applications
3. Built to support open standards like Open Tracing and Open Telemetry
4. Out of the box support for OCI developer and DB services
5. Native support for very large/high volume application
6. Superior price-performance

Learn more

oracle.com/manageability

Get informed
- Product news
- Customer videos
- Analyst reports

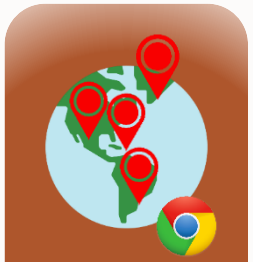
Get hands-on
- Free Tier/Trial
- Product demos
- Workshops

Engage with us
- Blogs
- Forums
- Webinars


ORACLE

Backup slides

APM Service Architecture



Metrics
HAR/Log
Screenshot

Traces/Spans/Metrics

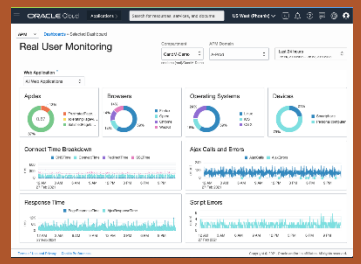
Traces/Spans

Application Performance Monitoring Service

OCI Tenant / Compartment/APM domain

APM User Interface

- Dashboards
- Trace Explorer
- Monitors
- RUM
- App Servers



Distributed Trace/Span Processor

Trace Data Store

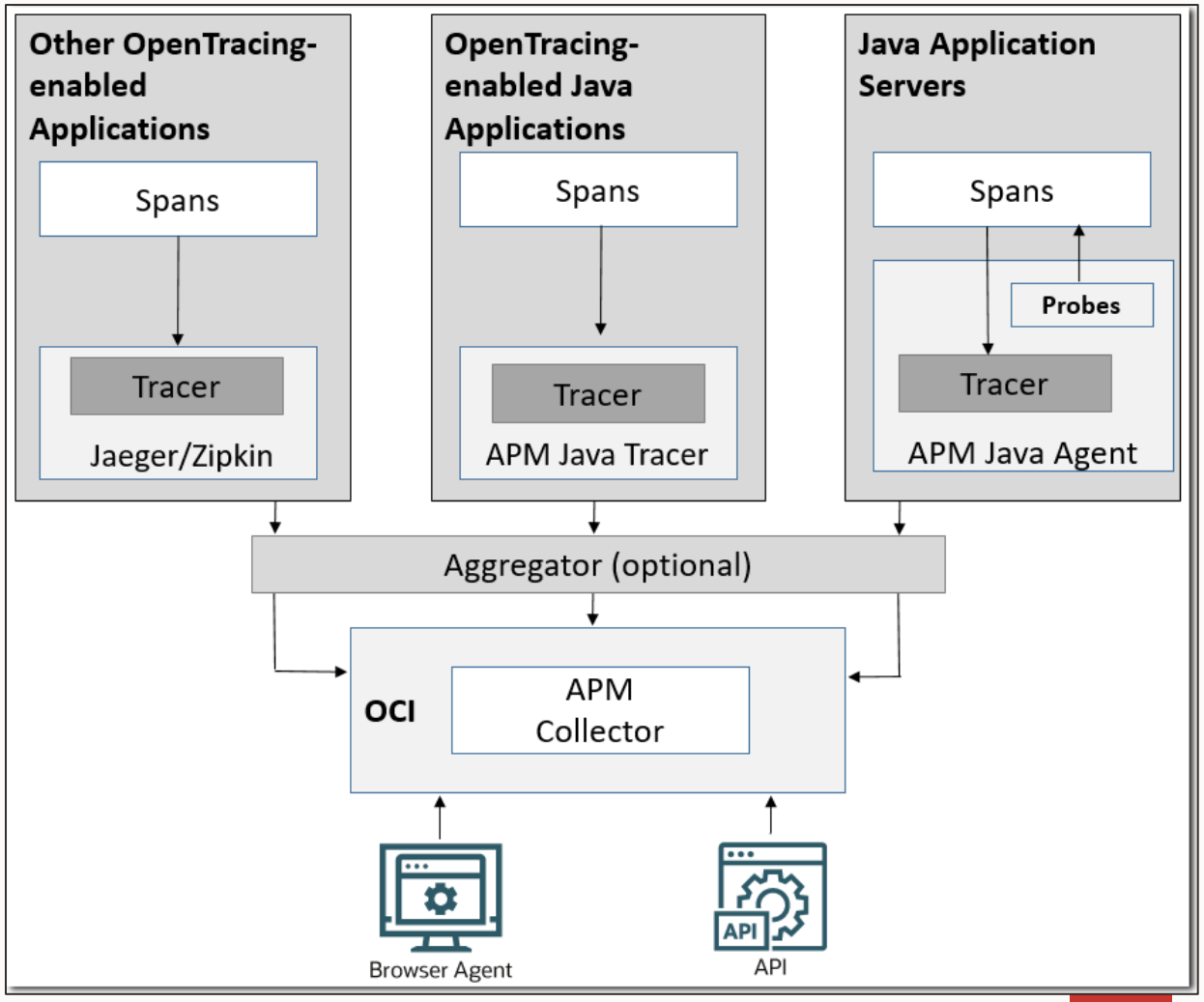
Monitoring

Notifications

APM Data Collector

APM Distributed Tracing – Data Sources/Deployment options

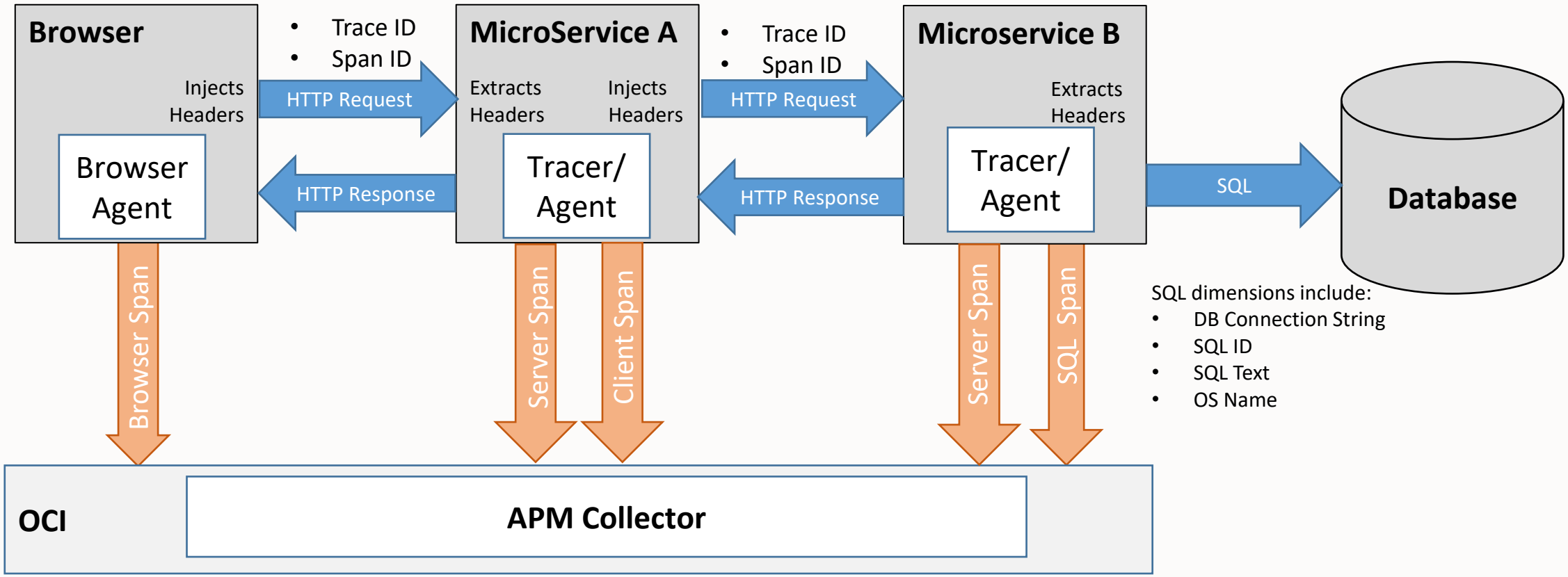
- **Java Agent**
  - Download agent from the cloud (with license/secret key private to the tenant)
  - Add to the monitored application using the java-agent parameter
- **Java Tracer**
  - Requires configuration and/or code change within the monitored application
- **Agent as aggregator**
  - Run the agent as a stand-alone process on the host (can be remote host)
  - Configure tracer/agent with the aggregator as the new end point
- **Browser Agent**
  - Javascript added manually to the application



Distributed Tracing, how does it work?

RUM dimensions include:
- HTTP Status
- OS Name
- Browser Type
- Connection Time
- Request URL
- Customer-defined Dimensions

Default dimensions in all spans:
- Service Name
- Operation Name
- Trace ID
- Span ID
- Start Time
- Duration
- Span Type-Specific Dimensions
- Customer-defined Dimensions



SQL dimensions include:
- DB Connection String
- SQL ID
- SQL Text
- OS Name