Agentic AI for Scalable Observability in OpenShift

Version 1.0 – July 31, 2025

**System Requirements**

**PaaS Solutions**

**oseAI**

VERSION: 1.0 REVISION DATE: July 31, 2025

Approval of the System Requirements indicates an understanding of the purpose and content described in this deliverable.

|  |  |  |  |
| --- | --- | --- | --- |
| **Approver Name** | **Title** | **Signature** | **Date** |
| Vikas Salvi | Sr IT Manager |  |  |
| Rajagopal Varadharajan | Principle IT Systems Engineer |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Section 1 Purpose 3](#_Toc198637000)

[Section 2 General System Requirements 3](#_Toc198637001)

[2.1 Major System Capabilities 3](#_Toc198637002)

[2.2 Major System Conditions 4](#_Toc198637003)

[2.3 System Interfaces 4](#_Toc198637004)

[2.4 System User Characteristics 4](#_Toc198637005)

[Section 3 Policy and Regulation Requirements 4](#_Toc198637006)

[3.1 Policy Requirements 4](#_Toc198637007)

[3.2 Regulation Requirements 4](#_Toc198637008)

[Section 4 Security Requirements 4](#_Toc198637009)

[Section 5 Training Requirements 4](#_Toc198637010)

[Section 6 Initial Capacity Requirements 5](#_Toc198637011)

[Section 7 Initial System Architecture 6](#_Toc198637012)

[Section 8 System Acceptance Criteria 6](#_Toc198637013)

[Section 9 Current System Analysis 6](#_Toc198637014)

[Section 10 References 7](#_Toc198637015)

[Section 11 Glossary 7](#_Toc198637016)

[Section 12 Document Revision History 7](#_Toc198637017)

[Section 13 Appendices 7](#_Toc198637018)

# Section 1 Purpose

The growing complexity of cloud-native, OpenShift-based microservices has elevated observability to a fundamental component of modern application management. As organizations scale workloads across dynamic clusters, traditional monitoring tools often fall short in addressing the rapid, transient characteristics of containers and distributed services. AI-driven observability solutions deliver advanced intelligence, enabling proactive anomaly detection, performance optimization, and automated remediation at scale.

This project examines the specific observability challenges present within OpenShift environments at Florida Blue, identifying critical monitoring areas and assessing the impact of AI-powered agents on Kubernetes workload management. The analysis aims to demonstrate how leveraging AI can transform operational efficiency and enhance the overall reliability of cloud-native systems.

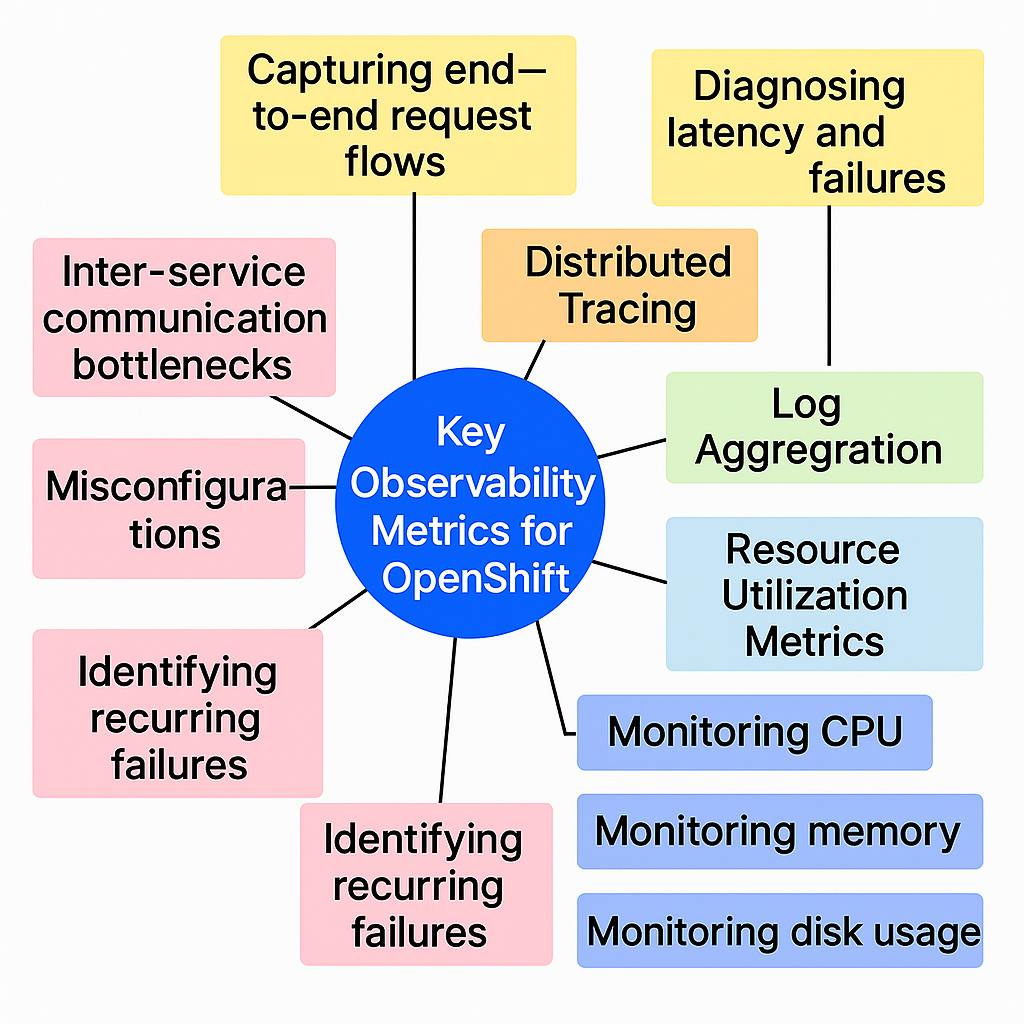


Figure 1. Observability Metrics

# Section 2 General System Requirements

## 2.1 Major System Capabilities

*Below are the major system capabilities in terms of availability, target deployment environment(s), device accessibility, and/or technical capability.*

* ***System must be available on the Internet for openshift-based deployments.***
* ***System must support high availability with failover mechanisms.***
* ***System must be deployable on multiple environments, including sandbox, development, production clusters.***
* ***System must be accessible via web and mobile interfaces.***
* ***System must support containerized applications.***
* ***System must integrate with CI/CD pipelines for automated deployments.***
* ***System must provide role-based access control (RBAC) for security management.***
* ***System must support monitoring and logging for performance tracking.***

## 2.2 Major System Conditions

*Assumptions and/or constraints (aka conditions). The conditions may limit the options available to the designer/developer.*

*- System must use Florida Blue OpenShift clusters*

*- System must use Florida Blue on-premise LLM models*

*- System must be compliant with Florida Blue mandated security policies*

*- System must be use Florida Blue exposed interfaces and services where applicable.*

## 2.3 System Interfaces

*Describe the dependency and relationship requirements of the system to other enterprise/external systems. Include any interface to a future system or one under development. For clarity, a graphical representation of the interfaces should be used when appropriate.*

## 2.4 System User Characteristics

*Identify each type of user of the system by function, location, and type of device. Specify the number of users in each group and the nature of their use of the system.*

# Section 3 The Importance of AI-Enhanced Observability in OpenShift

Kubernetes-based microservices introduce a level of complexity far beyond that of monolithic architectures. With countless interconnected components running across distributed platforms, ensuring effective observability becomes a significant challenge that traditional monitoring tools are often ill-equipped to address.

Key challenges include:

* **Scalability and Ephemerality**: Containers have short lifespans and scale rapidly, making it difficult to track ongoing patterns or recurring issues over time.
* **Dynamic Networking**: The frequent, on-the-fly communication between services complicates efforts to trace and diagnose inter-service requests.
* **Rapid Deployment Cycles**: The pace of continuous integration and delivery leads to frequent changes, increasing the chances that failures will go unnoticed.
* **Multi-Cluster and Hybrid Operations**: Running workloads across several clusters and virtual machines, and cloud services requires robust, centralized observability to maintain visibility and control.

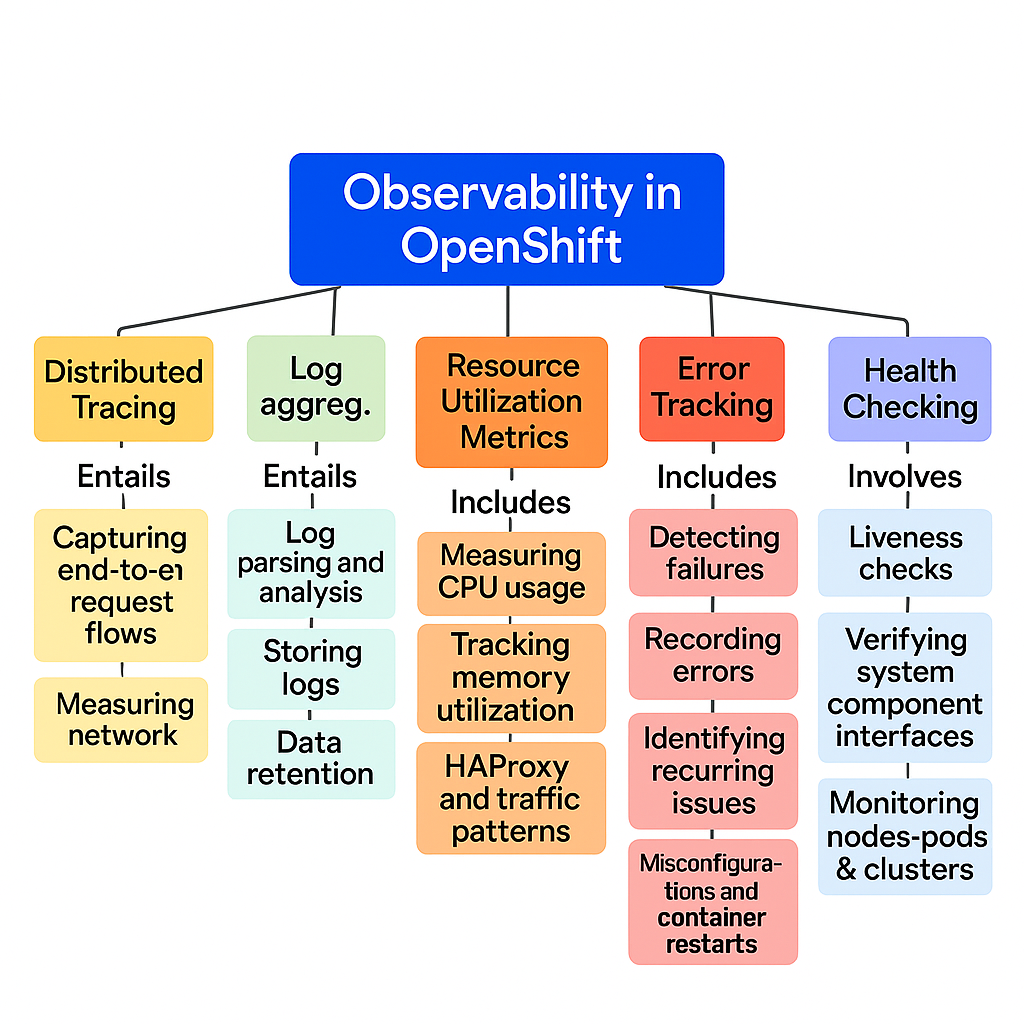
AI-driven observability solutions address these challenges by automating anomaly detection, streamlining performance optimization, and providing intelligent insights, empowering teams to manage complex Kubernetes environments with greater agility and confidence.

# Section 4 Essential Observability Metrics for OpenShift

Maintaining the reliability and performance of Kubernetes workloads requires a comprehensive approach to observability, with a focus on these key metrics:

* **Distributed Tracing**: Tracks end-to-end request paths across microservices, helping teams pinpoint sources of latency and identify points of failure.
* **Log Aggregation**: Gathers and analyzes logs from all containers and nodes, enabling the early detection of anomalies and incidents.
* **Resource Utilization**: Continuously monitors CPU, memory, and disk usage at the pod, node, and cluster levels to prevent resource bottlenecks and ensure optimal allocation.
* **Error Tracking**: Detects recurring issues, container restarts, and configuration errors, allowing for quicker remediation of persistent problems.
* **Network Telemetry**: Provides visibility into network traffic patterns, including ingress and egress flows, and highlights potential bottlenecks in inter-service communication.

Focusing on these metrics empowers teams to proactively manage OpenShift environments and optimize both system health and application performance.

   
 Figure 2. Key Observability Metrics

# Section 5 The Role of AI in Advancing OpenShift Observability

AI-enhanced observability takes monitoring beyond traditional, manual, and rule-based approaches by introducing automation and intelligence to Kubernetes environments. Solutions such as K8sAI bring several transformative benefits to day-to-day operations and incident management.

* **Proactive Anomaly Detection**: AI models continuously analyze historical data across metrics, logs, and traces. This enables the early detection of unusual patterns or deviations, allowing teams to address potential issues before they impact system reliability or user experience.
* **Automated Root Cause Analysis**: By correlating diverse signals—such as logs, traces, and resource metrics—AI-driven tools pinpoint the underlying causes of incidents. This results in faster, more accurate diagnosis and generates actionable recommendations for remediation.
* **Predictive Maintenance and Optimization**: Leveraging machine learning algorithms, solutions like K8sAI forecast possible failures or performance bottlenecks. Teams can then proactively scale resources or fine-tune configurations to prevent outages and maintain optimal performance.
* **AI-Powered Log Analysis and Insights**: oseAI seamlessly integrates with centralized logging platforms like Elasticsearch, Loki, or Fluentd. It enables users to submit natural language queries, such as “Why did pod X restart multiple times today?” The tool then analyzes relevant logs, uncovers patterns, and delivers targeted, actionable insights.

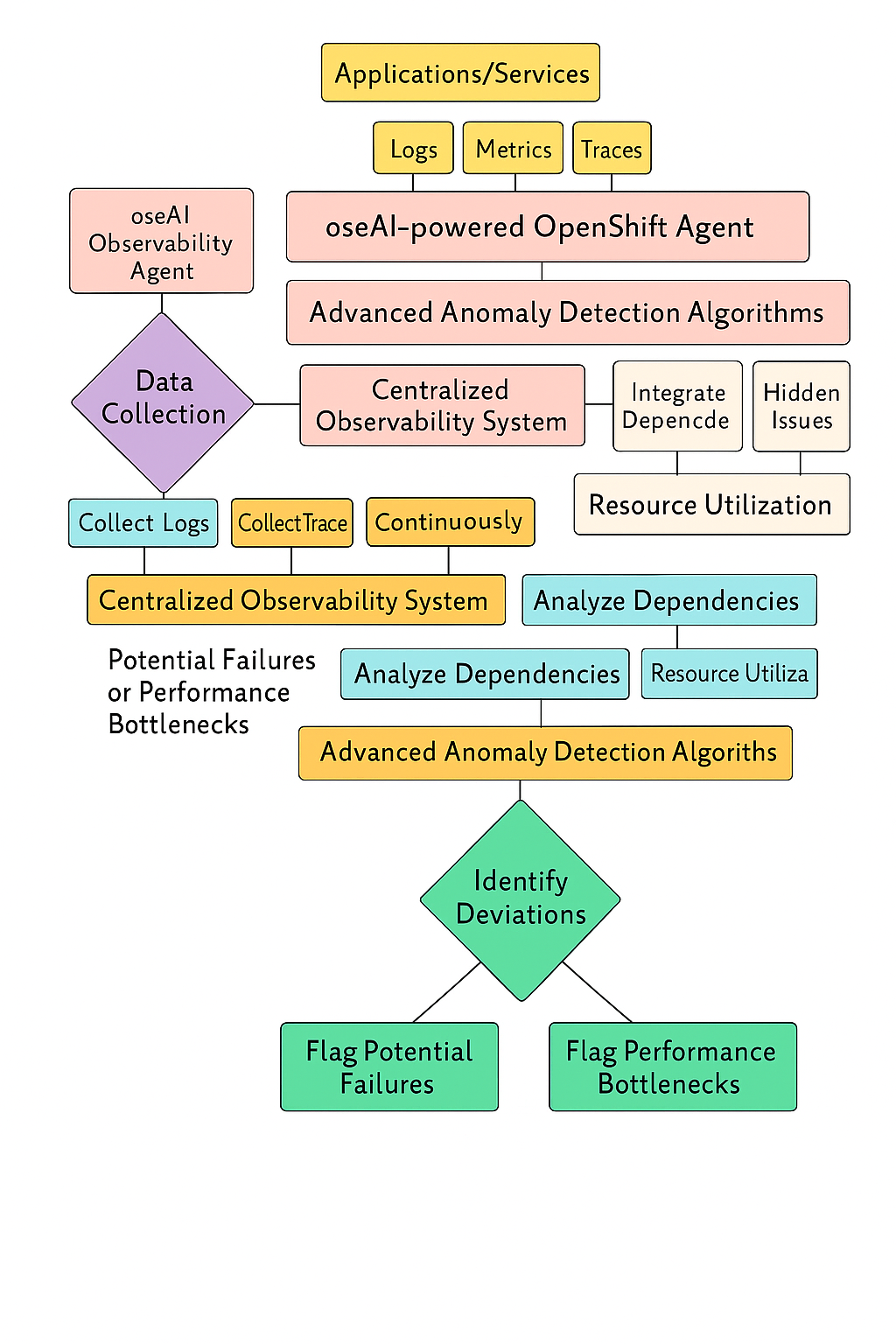
By implementing AI-driven observability, organizations gain a smarter, faster, and more adaptive approach to managing and maintaining complex Kubernetes workloads.

# Section 6 Automated Remediation and Self-Healing with {OSE} AI Agents

AI Agents empower OpenShift clusters to automatically respond to anomalies, reducing downtime and minimizing manual intervention. These systems use intelligent triggers to take swift, predefined actions when issues are detected.

* **Dynamic Scaling**: When CPU utilization surges beyond acceptable thresholds, K8sAI can automatically scale deployments to ensure continued performance and availability.
* **Pod Recovery**: If health checks indicate a pod has failed, the platform initiates an immediate restart to restore service without the need for operator involvement.
* **Automated Rollbacks**: In the event of elevated error rates, K8sAI can execute a rollback to the previous stable deployment, quickly mitigating the impact of faulty releases.

This self-healing approach not only enhances system reliability but also allows teams to focus more on innovation rather than routine maintenance and firefighting.

 Figure 3. OpenShift Agentic AI Framework

# Section 7 AI Agents Architecture

*hardware, software, programming languages, tools and operating system requirements for the application or project.*

1. *OpenShift*
2. *VSCode*
3. *Golang, Python, scripts, NodeJS  
     
   Picture ???*

# Section 8 AI Agents List and Description

*List AI Agents*

*???*

# Section 9 Agentic AI UI/Front-end

*Describe UI or frontend used to test AI Agents*

*???*

# Section 10 System Acceptance Criteria

*Describe UI or frontend used to test AI Agents*

*???*

# Section 10 References

*List of all documents and other sources of information referenced in this document and utilized in its development.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Document No.** | **Document Title** | **Date** | **Author** |
|  |  |  |  |
|  |  |  |  |

# Section 11 Glossary

*Acronyms required to properly interpret the requirements contained within this document.*

# Section 12 Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
|  |  |  |  |
|  |  |  |  |

# Section 13 Appendices