High-Level Design Document

for

Data and Authentication Monitoring Platform (DAMP)

[1. Introduction 1](#_Toc151663627)

[2. System Architecture 2](#_Toc151663628)

[2.1. Monitoring Components 2](#_Toc151663631)

[2.2. Scalability 2](#_Toc151663632)

[2.3. Self-Diagnosis and Notification 2](#_Toc151663633)

[3. Security Considerations 4](#_Toc151663634)

[3.1. Data Protection 4](#_Toc151663636)

[3.2. Access Controls 4](#_Toc151663637)

[4. Cost-Effective Design 5](#_Toc151663638)

[4.1. Cloud Resource Optimization 5](#_Toc151663639)

[4.2. Open-Source Tools 5](#_Toc151663640)

[5. Documentation and Training 6](#_Toc151663641)

[5.1. Comprehensive Documentation 6](#_Toc151663642)

[5.2. Training for Operational Support 6](#_Toc151663643)

[6. Continuous Improvement 7](#_Toc151663644)

[6.1. Regular Reviews 7](#_Toc151663646)

[6.2. Feedback Mechanism 7](#_Toc151663647)

[7. Conclusion 8](#_Toc151663648)

# Introduction

The Data and Authentication platform has multiple distributed systems that need to be proactively monitored. These systems host critical services utilising multiple technologies and running variety of applications which generate stats, metrics and logs.

The monitoring system the platform is designed to ensure continuous availability, security, and optimal performance. It encompasses a diverse set of components for collecting, analysing, and visualising metrics and logs from the multiple distributed systems on the platform.

# System Architecture



# Monitoring Components

* + 1. Monitoring Agents
       1. Deploy on each system for real-time data collection.
       2. Collect metrics and logs for both the operating system and services/applications.
    2. Centralized Monitoring Server

This will be provisioned with:

* + - 1. Prometheus for metrics and alerting.
      2. ELK Stack (Elasticsearch, Logstash, Kibana) for log management and visualisation.
      3. Grafana for creating dashboards and visualization.
    1. Security Layer
       1. Encrypt data in transit and at rest.
       2. Implement strong authentication mechanisms for access control.
    2. Incident Management Integration
       1. Integration with an incident management system like PagerDuty for immediate alert notification and response coordination by the operational team.

# Scalability

* + 1. Horizontal Scaling

Monitoring components will be deployed to be scalable horizontally handle growing infrastructure.

* + 1. Container Orchestration

To facilitate dynamic scaling of applications hosted on the platform, container orchestration like Kubernetes will be leveraged.

* + 1. Database Scalability

Scalable databases to be implemented for storing metrics and logs.

* + 1. Load Balancing

To ensure high availability and efficient distribution of monitoring tasks.

# Self-Diagnosis and Notification

* + 1. Anomaly Detection

Anomaly detection algorithms will be implemented in Prometheus to identify deviations from normal behaviour and annunciate such anomalies within the system.

* + 1. Automated Alerts

Automated alert will be configured based on predefined thresholds and anomaly detection.

* + 1. Self-Healing Mechanisms

Self-healing mechanisms like automated service restarts or resource scaling will be implemented where possible.

# Security Considerations



# Data Protection

* + 1. Personal Data Handling
       1. Anonymise personal data in logs to protect privacy.
       2. Implement access controls to restrict access to sensitive monitoring data.
    2. Encryption
       1. Use TLS for encrypting data in transit.
       2. Encrypt stored data, especially logs containing personal information.

# Access Controls

* + 1. Role-Based Access Control (RBAC)
       1. Implement RBAC to control access to monitoring data based on roles and responsibilities.
    2. Audit Logs
       1. Generate audit logs to track access and changes to the monitoring system.

# Cost-Effective Design

# Cloud Resource Optimization

* + 1. Reserved Instances
       1. Utilize reserved instances or spot instances to optimize costs in hybrid cloud deployments.
    2. Resource Scaling
       1. Implement auto-scaling to dynamically adjust resources based on demand, minimizing unnecessary costs.

# Open-Source Tools

* + 1. Open-Source Monitoring Tools
       1. Leverage open-source monitoring tools (Prometheus, Grafana, ELK Stack) to reduce licensing costs.
    2. Cost-Aware Configurations
       1. Optimize configurations to reduce resource usage and associated costs.

# Documentation and Training

# Comprehensive Documentation

* + 1. Maintain detailed documentation for installation, configuration, and troubleshooting.

# Training for Operational Support

* + 1. Provide training sessions for the Operational Support team on monitoring tool usage and incident response procedures.

# Continuous Improvement



# Regular Reviews

* + 1. Conduct regular reviews of the monitoring system to identify areas for improvement.

# Feedback Mechanism

* + 1. Establish a feedback mechanism with the Operational Support team to incorporate their experiences and insights.

# Conclusion

The implementation of the design items outlined in this document will make for a highly available and robust monitoring system that ensures the continuous health and security of the Data and Network Authentication platform. Regular updates and continuous improvement practices will contribute to the system's long-term effectiveness.