Deployment and Monitoring Guide for Gen-Al Enabled Integrated Platform Environment

1. Deployment Steps

Follow the steps below to deploy and configure the Gen-Al Enabled Integrated Platform Environment (IPE) on your infrastructure.

1.1 Clone Repository

Clone the GitHub repository that contains the source code for both the frontend (React) and backend (Python) systems. Ensure that all submodules and dependencies are included during cloning.

GitHub Repository: https://github.com/ewfx/gaipl-mind-over-machines

1.2 Set Up Project Environment

Create a new project environment for deployment, or use an existing environment. Make sure that necessary services are activated, such as OpenShift, ElasticSearch, and other integrations for monitoring and incident management.

1.3 Build Docker Images

Build Docker images for both the frontend and backend systems. Tag the images appropriately for deployment.

Command: gcloud builds submit --tag gcr.io/your-project-id/genal-platform

1.4 Push Images to OpenShift Container Registry

Push the built Docker images to your OpenShift Container Registry. Ensure that you have proper permissions and authentication.

1.5 Deploy Frontend and Backend

Deploy the React frontend and Python backend to OpenShift. Configure environment variables, such as API keys, credentials, and storage locations.

1.6 Deploy AI Model and Integration

Deploy the Llama70B AI model and integrate it with the backend. Ensure proper environment variables for the AI model location and configurations.

1.7 Configure Database (ElasticSearch)

Set up and configure ElasticSearch for real-time indexing and searching of incident data. Ensure proper cluster configuration and data indexing.

1.8 Set up API and WebSockets

Configure the API and WebSocket connections between the frontend, backend, and incident management tools. This ensures that the platform supports real-time communication.

2. Monitoring Guide

Use the following monitoring guidelines to ensure the platform's performance, stability, and security after deployment.

2.1 Monitor Platform Health

Use OpenShift's built-in monitoring tools or third-party integrations to monitor the system's CPU, memory, and disk I/O usage. Set up automatic scaling to handle high resource consumption.

2.2 Track Real-Time Metrics

Monitor real-time metrics such as incident resolution rate, open incidents, and platform health status. This data is crucial for ensuring smooth operations.

2.3 Log Monitoring and Error Tracking

Track logs generated by the backend, frontend, and AI models. Use ElasticSearch to store and analyze logs for error tracking.

2.4 Alerting and Notification Setup

Set up alerting mechanisms to notify administrators of any system failures or anomalies, using tools like Stackdriver or OpenShift alerts.

2.5 Model Performance Monitoring

Monitor the performance of the AI model (Llama70B) in the Prediction system. Track metrics such as accuracy, precision, and recall.

2.6 Continuous Improvement and Optimization

Review monitoring data and logs regularly to identify opportunities for improvement. Use insights to optimize the platform and the AI model.

3. Post-Deployment Tasks

3.1 Documentation and Knowledge Sharing

Document all deployment procedures, configurations, and monitoring setups. Share knowledge with the team to ensure proper maintenance and troubleshooting.

3.2 Backup and Disaster Recovery

Ensure that backup strategies are in place for critical data and configurations. Regularly test disaster recovery procedures.

3.3 Security Auditing and Compliance

Conduct regular security audits and vulnerability assessments. Ensure compliance with relevant regulations and industry standards.

3.4 Scaling and Resource Optimization

Monitor system performance and scale resources as needed to handle increased load. Optimize resource utilization to reduce costs.