Proposal: Unified Performance Analysis and Reporting Platform

Objective

To develop a Spring Boot + React based solution that serves as a one-stop platform for performance analysis, monitoring, and reporting across multiple environments (Azure VMs with Docker Compose, Docker Swarm, and AKS clusters).

Key Features

1. Environment & Version Selection
- Environment Dropdown: Select from multiple environments (Azure VMs, Docker Swarm, AKS).
- Simulator Version Dropdown: Choose specific application release/simulator version.
- Dynamic Dashboard: Loads performance data and service status based on chosen environment and version.
2. Real-Time Dashboard
- Service Monitoring
- Status of each running service (Running / Stopped / Paused).
- Memory consumption and resource utilization metrics.
- Control Panel
- Start, Stop, Pause, Restart options for services.
- Secure API integration to trigger container lifecycle actions.
- Graphs & Metrics
- Real-time memory/CPU utilization.
- Request throughput and latency.
- Log-derived performance insights.
3. Log Data Collection & Analysis
- Log Parsing: Extract key performance indicators (errors, latency, throughput, memory usage).
- Centralized Storage: Logs and metrics stored in a persistent database for analysis.
- Visualization: Time-series charts, trend graphs, and error distribution analysis.
4. Historical Data & Reporting
- Historic View: Select past runs by environment and version.
- Comparison Reports: Compare multiple runs across releases and environments.
- Export Options: Download performance reports in PDF/Excel format.

Architecture Overview

Frontend (React)
- Interactive dashboard with charts and controls.
- Secure user authentication and role-based access.
Backend (Spring Boot)
- REST APIs for environment/service management.
- Log collection and parsing service.
- Database integration for storing historical data.
- APIs for service lifecycle operations (start/stop/restart).
Data Storage
- Relational Database (Postgres/MySQL) for structured data.
- Time-series database (optional) for performance metrics.
Infrastructure Integration
- Works across Azure VMs (Docker Compose), Docker Swarm, and AKS clusters.

- Unified API layer to abstract different environments.

Benefits

- Single Pane of Glass: One platform for monitoring multiple environments.
- Improved Efficiency: Reduce manual efforts in performance test monitoring.
- Proactive Troubleshooting: Real-time insights and quick service actions.
- Historical Tracking: Informed decision-making based on past trends.
- Scalable & Extensible: Future support for additional environments and metrics.

Next Steps

1. Finalize requirements with stakeholders.
2. Define MVP scope (basic monitoring + service control).
3. Develop backend APIs and frontend dashboard.
4. Integrate with logging systems and databases.
5. Pilot run in one environment (Azure VMs).
6. Rollout to Docker Swarm and AKS clusters.

Visual Mockup (Conceptual)

- Dropdowns: Environment & Version selection at top.
- Dashboard Layout:
- Left panel: List of services with status & action buttons.
- Right panel: Graphs for CPU/Memory/Latency.
- Bottom panel: Logs & alerts.
- Historic View: Tab to switch from 'Current' to 'Historic' reports.

Conclusion

This platform will streamline performance monitoring, provide actionable insights, and empower teams with better control over their services across environments.