Phase 2 Documentation: ADF Diagnostic Logging to Log Analytics

# 🧾 Project Title:

SmartAPIStreamX – Phase 2: Monitoring Enablement with Log Analytics Integration

# 🎯 Objective:

To enable real-time diagnostic logging in Azure Data Factory (ADF) by integrating it with Azure Log Analytics, forming the foundation for AI-driven monitoring, error detection, and self-healing pipelines.

# ✅ Key Outcomes:

• ADF linked with Log Analytics workspace

• Diagnostic logs collected: PipelineRuns, ActivityRuns, TriggerRuns

• Logs queryable via KQL (Kusto Query Language)

• Ready for AI-based alerting, dashboards, and auto-remediation

# 📦 Tools & Services Used:

Azure Data Factory – Data pipeline orchestration

Azure Log Analytics – Log storage & querying engine (KQL)

Azure Monitor – Centralized monitoring and diagnostic engine

Microsoft.Insights – Resource provider for diagnostic settings

# 🔧 Step-by-Step Implementation

## 1️⃣ Enable Resource Provider

• Navigated to: Azure Portal → Subscriptions  
• Selected active subscription → Resource Providers  
• Registered: Microsoft.Insights (required for diagnostics)

## 2️⃣ Create Log Analytics Workspace

• Azure Portal → Create Resource → Log Analytics Workspace  
• Defined:  
 - Name: SmartAPILogAnalytics  
 - Resource Group: SmartAPI-ADF-RG  
 - Region: Central India  
• Created workspace successfully

## 3️⃣ Enable Diagnostic Settings in ADF

• Opened: Azure Data Factory (e.g., datafactoryparesh)  
• Navigated: Monitoring → Diagnostic Settings  
• Clicked: ➕ Add Diagnostic Setting  
• Configuration:  
 - Name: ADF-Logs-To-LogAnalytics  
 - Log Categories Enabled:  
 ✅ PipelineRuns  
 ✅ ActivityRuns  
 ✅ TriggerRuns  
 - Destination: ✅ Send to Log Analytics  
 - Selected: SmartAPILogAnalytics  
• Saved successfully after Microsoft.Insights registration

# 🧪 Example KQL Query:

ADFActivityRun  
| where Status == "Failed"  
| project TimeGenerated, PipelineName, ActivityName, Error, FailureType  
| order by TimeGenerated desc

# 📈 Result:

• Logs now stream from ADF into Log Analytics  
• Errors are captured in near real-time  
• Foundation ready for:  
 - Power BI dashboards  
 - GPT-based RCA/fix recommendations  
 - Automated alerts & retry workflows

# 📁 Assets Created:

• Azure Data Factory: datafactoryparesh  
• Log Analytics Workspace: SmartAPILogAnalytics  
• Diagnostic Setting: ADF-Logs-To-LogAnalytics  
• Subscription Provider: Microsoft.Insights

# 💡 Advantages by Service:

## ✔ Azure Data Factory:

• Centralized pipeline management and orchestration  
• Built-in support for logging and monitoring  
• Scalable across multiple data sources

## ✔ Azure Log Analytics:

• Powerful KQL queries for analyzing diagnostics  
• Supports integration with alerts and dashboards  
• Enables real-time monitoring of data workflows

## ✔ Microsoft.Insights:

• Enables diagnostic logging for ADF  
• Integrates with Azure Monitor and Log Analytics  
• Essential for enabling observability

## ✔ Azure Monitor:

• Consolidates logs, metrics, and alerts  
• Facilitates proactive monitoring  
• Supports visualizations, dashboards, and autoscale

# Enhancements Added to Phase 2

## 1. GPT Integration for Fix Recommendations

• Integrated Azure OpenAI (GPT) API with Log Analytics error output for dynamic Root Cause Analysis (RCA) and fix suggestions.  
• Logs with status == "Failed" are analyzed using GPT prompt chaining to provide contextual solutions.  
• Responses are displayed in the dashboard or routed to notifications (Email, Teams).

### Advantages for the Organization:

✔ Faster resolution of pipeline failures with intelligent suggestions.  
✔ Reduces manual troubleshooting and accelerates recovery.  
✔ Leverages organizational error history to train GPT prompts for accuracy.

### Key Features & How It Works:

1. Extract failed logs using KQL in Log Analytics.  
2. Send error string to Azure OpenAI using Logic App or Function App.  
3. Receive RCA + Fix.  
4. Use in dashboard, email, or automation logic.

## 2. Alert System (Logic App + Email)

• Azure Logic App triggers on ADF failure events (via Event Grid).  
• Parses error, formats HTML alert email.  
• Sends notifications via Outlook or SendGrid with actionable insights.

### Advantages for the Organization:

✔ Real-time alerting with full context of failure.  
✔ Improves SLA response and engineering efficiency.  
✔ Customizable templates and alert thresholds.

### Key Features & How It Works:

1. Logic App listens to pipeline failure event.  
2. Parses ADF metadata (Pipeline, Activity, Error).  
3. Formats HTML using built-in Compose and Send Email.

## 3. Retry / Self-Heal Logic

• ADF native retry policy used for transient errors with exponential backoff.  
• For fatal errors, Logic App queues the re-run of failed activity via ADF REST API.  
• Optional: Update watermark to resume from last known good state.

### Advantages for the Organization:

✔ Increased reliability and resiliency of data pipelines.  
✔ Reduces manual intervention and escalations.  
✔ Enhances SLA compliance and pipeline availability.

### Key Features & How It Works:

1. Configure retry policy in Copy/Lookup Activity (exponential retry).  
2. Use Logic App to re-trigger ADF pipeline using stored state (watermark).  
3. Integrate with monitoring dashboard for visibility.