Hewlett Packard
Enterprise

HPCM MONITORING PIPELINE VISUALIZATION

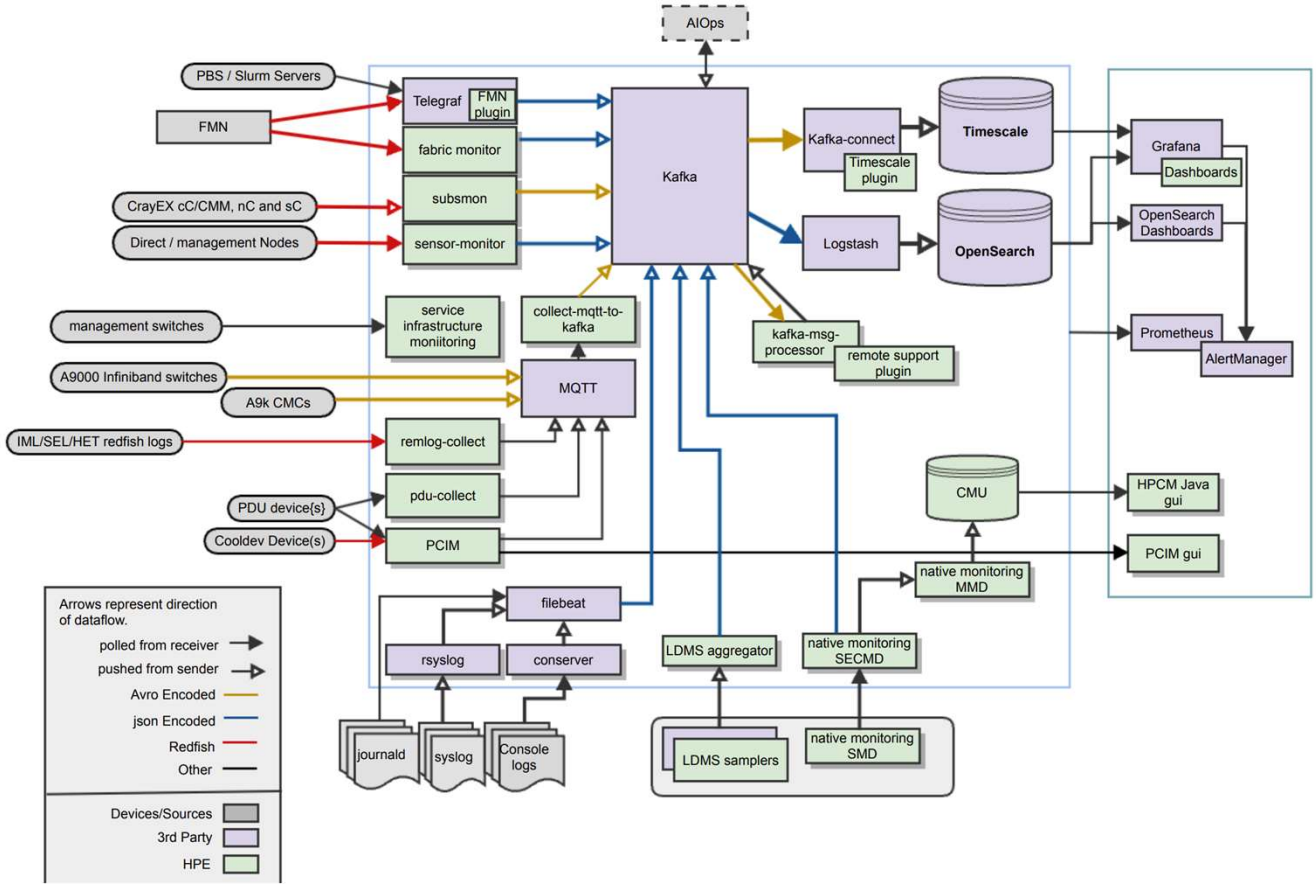
Vidyavardhaka College Of Engineering

CTY Members – Akshay G , Chandana L, Chinmai H K , Abhilash T R , Srivarshini S

Raghul Vasudevan

March 12 2024

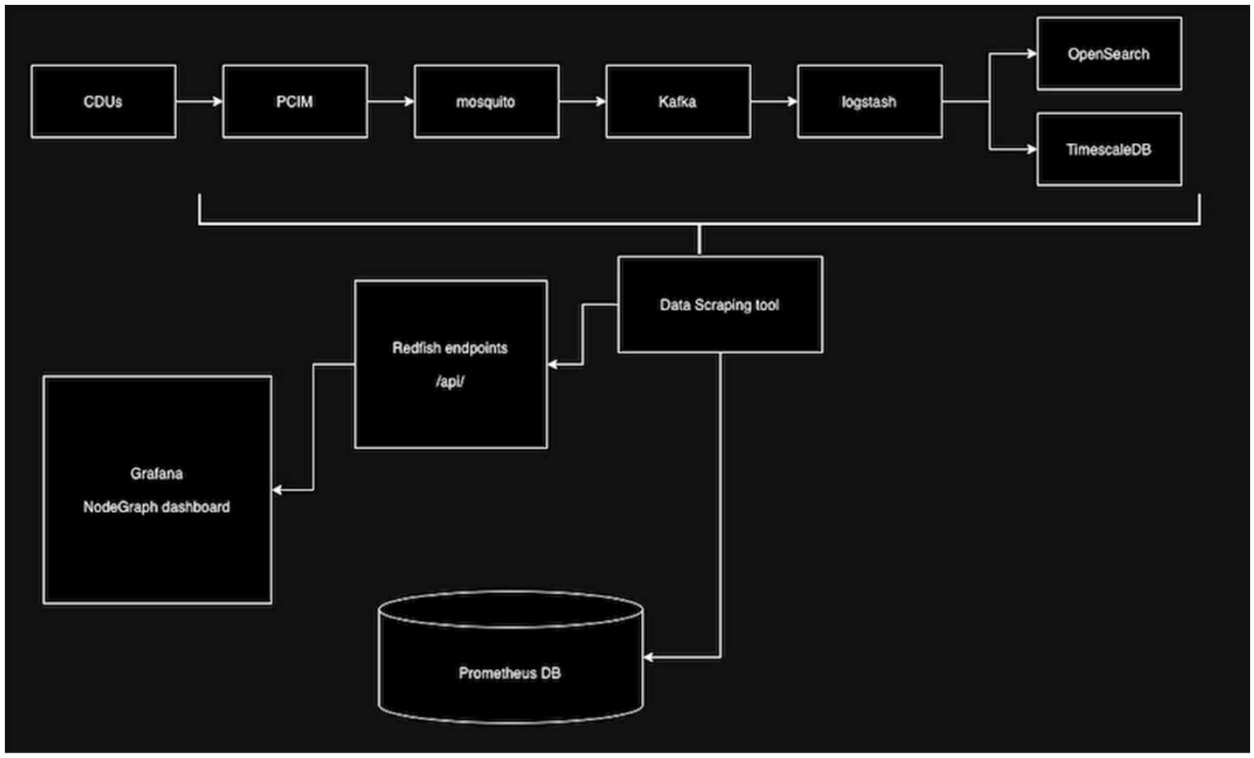
PROBLEM STATEMENT



- Currently, the user must run different set of commands or mon service status check to debug the respective monitoring pipeline to find the issues.
- The end user/customer finds difficulty in understanding the HPCM monitoring stack and pipelines. (including Support and on-site folks).
- The purpose of this project is to reduce the difficulty in debugging the monitoring stack/components/services and pipelines for the customer and support team.

PROJECT ARCHITECTURE, OBJECTIVES AND SKILL

Architecture (Diagram below)



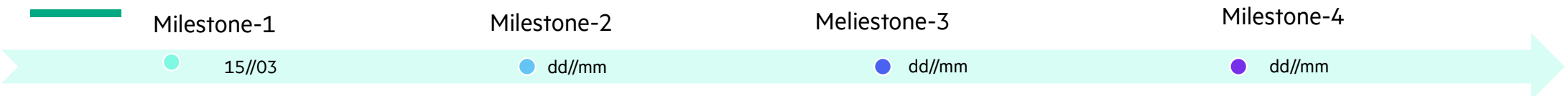
Objectives

1. Data scraping from components of the pipeline.
2. Real-time monitoring.
3. Visualization of status of the pipeline.

Skill Set

1. Programming language - Python
2. Grafana : Nodegraph API Plugin
3. Linux Operating Environment
4. RedFish End-Point

PROJECT TIMELINE , MILESTONE , STATUS



Milestone	Milestone Description	Tasks	Status	Next Steps / Issues
Milestone-1	Initial POC with one sample pipeline and minimal metrics	1. Investigation 2. Design 3. Data Collection / Scrapping tool 4. Data Persistence method 5. Grafana NodeGraph dashboard	1. Completed the investigation on the data scrapping tool from the required components and written a method for data persistence and created a sample POC dashboard with NodeGraph.	Next: 1. Explore more on data persistence method with NodeGraph plugin 2. Further pipelines implementation Issues: • HTTP server hosting for data scrapping • DB to Grafana NodeGraph visualization
Milestone-2				
Milestone-3				
Milestone-4				

AGENDA

- About the Identified monitoring pipeline – CDU and Schema design
- Data collection / scrapping – components involved in pipeline
- Data persistence
- Grafana NodeGraph dashboard
- Demo

SCHEMA DESIGN – CDU MONITORING PIPELINE

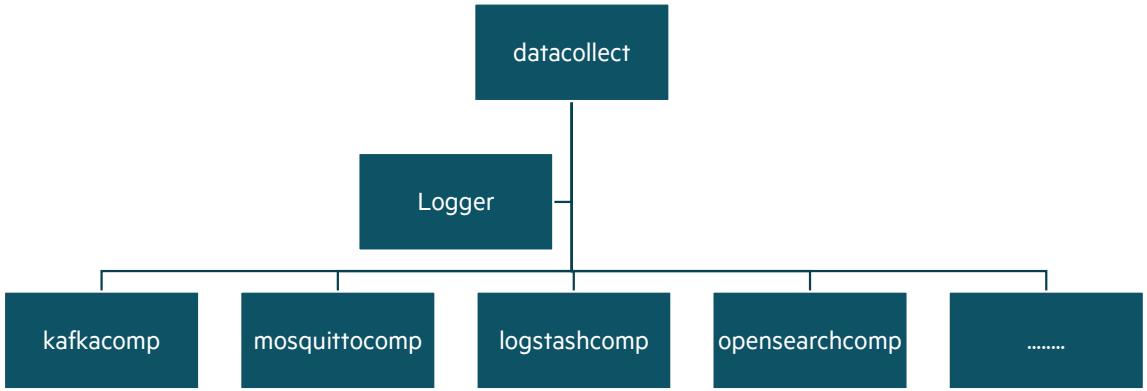
- Chosen metrics has been added in the data collection tool for POC.
- Further metrics will be added for every components later.

SERVICES	STATUS	UPTIME	DATA FLOW
PCIM	Running	Since 2 days	0/1
Mosquitto	Active/Inactive	24hrs	0/1
Kafka	Active/Inactive	Since 10 days, 23hrs	0/1
Logstash	Active/Inactive	Since 12 days, 10hrs	0/1
Opensearch	Active/Inactive	Since 25 days, 5hrs	0/1

DATA COLLECTION/ SCRAPPING

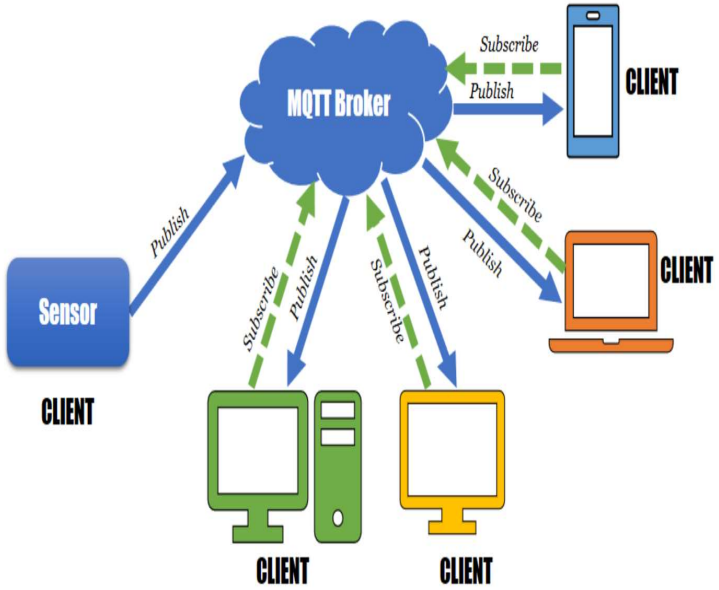
- Mosquitto
- Apache Kafka
- Logstash
- OpenSearch

- Installed the above services and configured to run it as a service
- Extracted the status and uptime using the subprocess module of python
- The extracted status is inserted into a redfish endpoint
- This process is done every 5 seconds to fetch us a time series data structure.

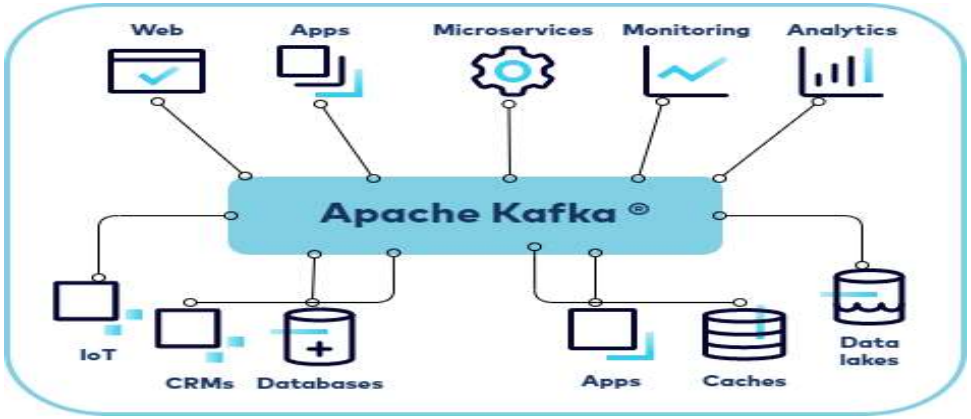


ABOUT COMPONENTS

Mosquitto

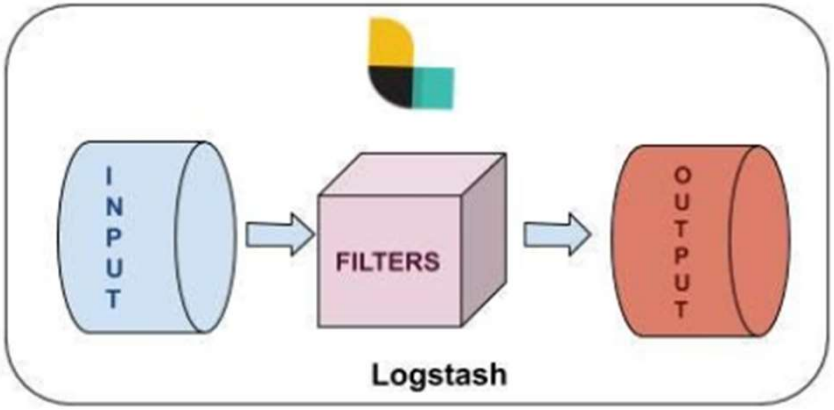


Apache Kafka

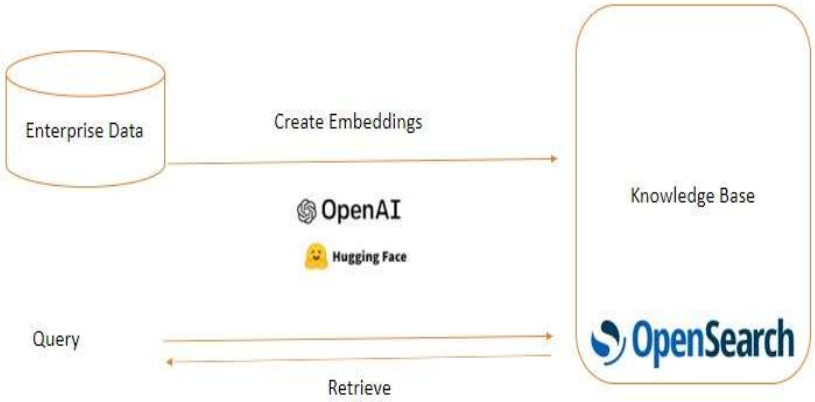


DATA COLLECTION/ SCRAPPING

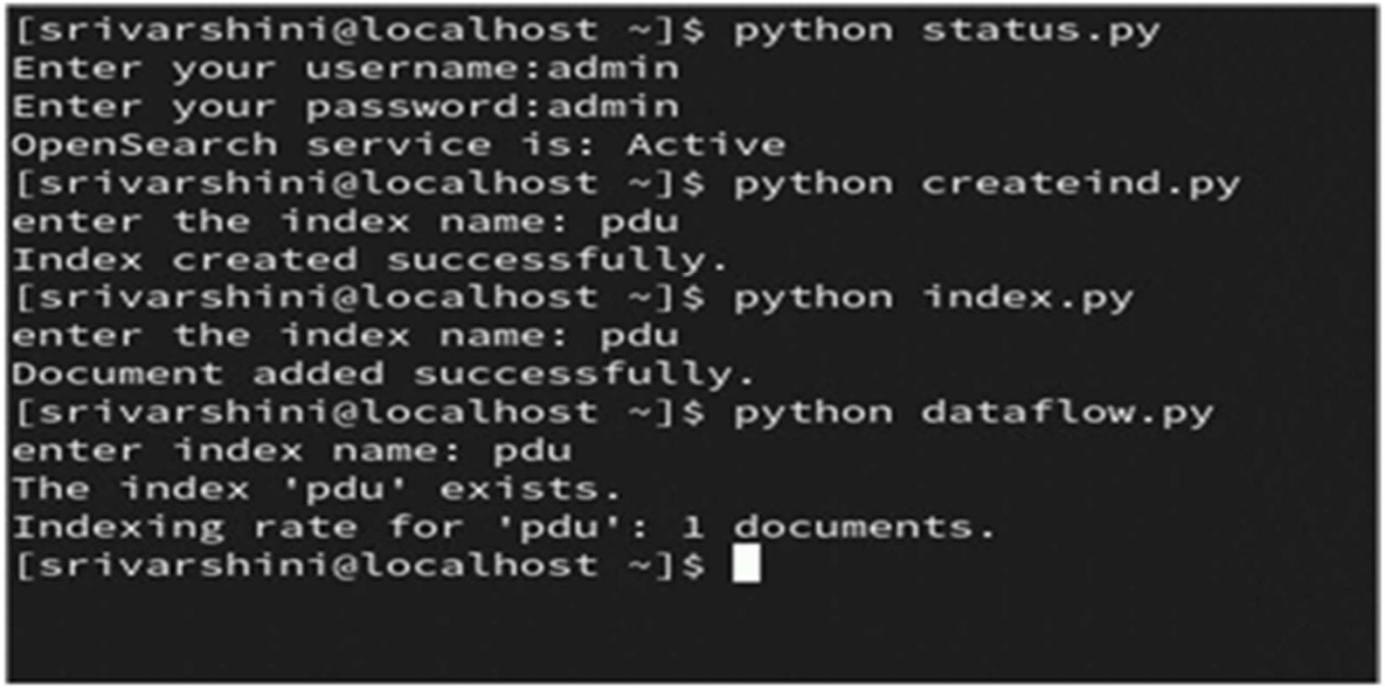
Logstash



OpenSearch

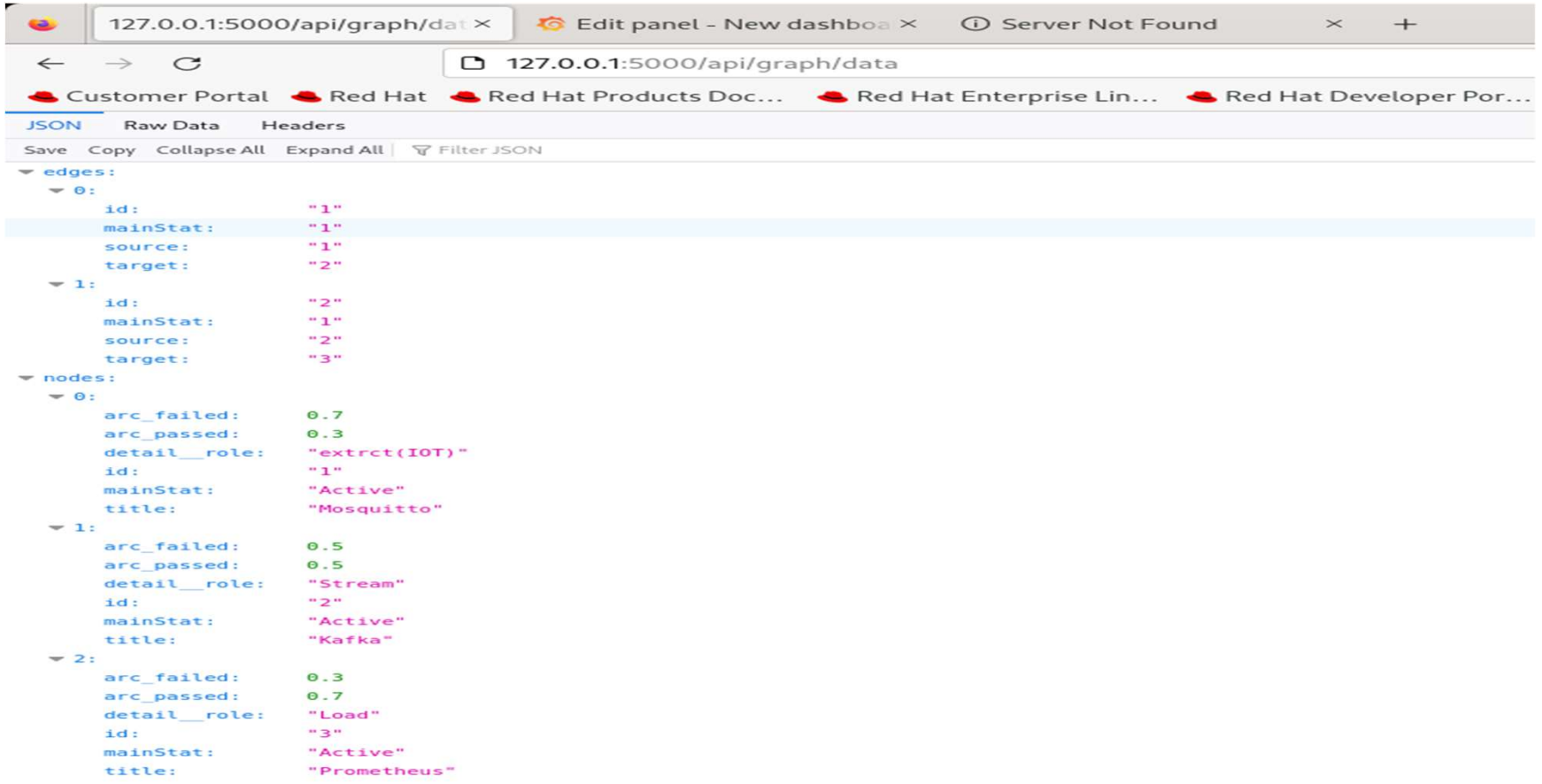


OPENSEARCH DATA FLOW CHECK



OpenSearch service status has been extracted and an index named pdu has been created and data has been injected into it dataflow for that index has been verified

DATA PERSISTENCE – API SERVER



GRAFANA - NODEGRAPH

- Grafana Node Graph is a plugin for Grafana that provides a specialized visualization panel for displaying interconnected nodes and edges.
- Grafana Node Graph API, on the other hand, is a separate component or functionality that allows interaction with the Grafana Node Graph plugin programmatically through an API interface.
- Node Graph API is an API interface that allows programmatic interaction with the node graph visualization.

DEMO AND NEXT STEPS

Demo Objective

1. How status is collected presently
2. Executing the code
3. Presenting the API endpoint
4. Grafana- Visualization of services

Next Steps

1. Checking the status of data flow between the components.
2. Integrating more pipelines.
3. Multiple pipelines visualization.

PROJECT PLAN , LEARNINGS , CHALLENGES

Learnings

1. **Linux Operating Environment**
2. **Components – Kafka, Mosquitto, Logstash, Opensearch**
3. **Prometheus DB**
4. **Grafana - Dashboard**

Challenges

1. **Prometheus**
2. **Nodegraph vs API plugin**
3. **API Endpoints – Multiple and Single**

THANK YOU