# Real-Time Distributed Air Quality Monitoring System End-to-End Big Data Pipeline Project

Presented by:

Mujahid Afzal

Muhammad Abdullah Zia

Talha Siddique

# Problem Statement & Objective: Breathing New Life into Air Quality Monitoring

**The Problem:** Traditional air quality reports are often static and quickly outdated, failing to capture the dynamic nature of urban air pollution. This lag in information can hinder timely interventions and public awareness.

**Our Solution:** We propose a streaming pipeline designed to provide instant visibility into critical particulate matter (PM2.5, PM10) and other environmental factors.

1. **Ingest Live Data**

   From the OpenAQ API.

2. **Distributed Processing**

   Using a scalable cluster.

3. **Dual-Storage Strategy**

   For both real-time and archival data needs.

# High-Level Architecture: The Data Flow Blueprint

1

## Source: OpenAQ API

Live sensor data from around the globe.

2

## Ingestion: Apache Kafka

Robust and scalable data buffering.

3

## Processing: Apache Spark Structured Streaming

Real-time data transformation and analysis.

4

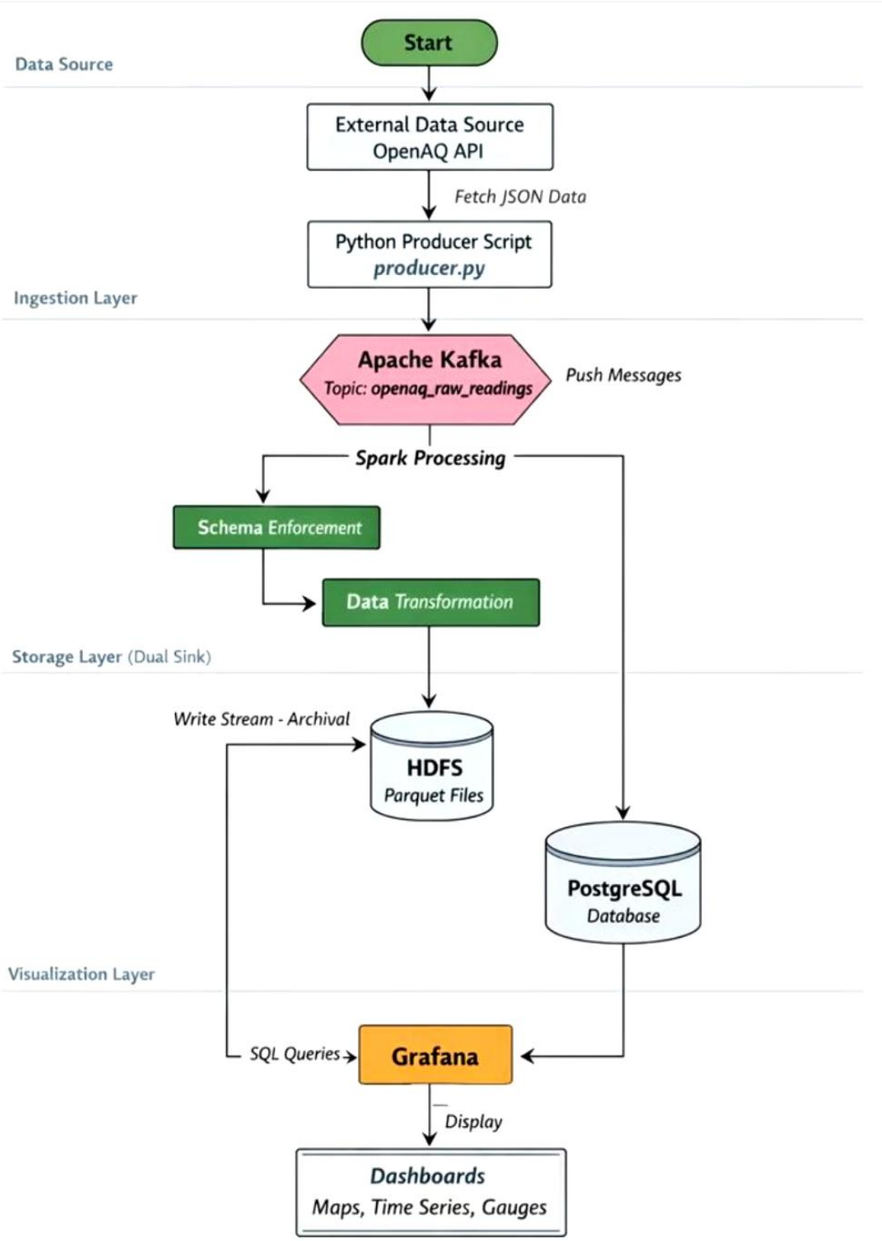
## Storage: HDFS & PostgreSQL

Long-term archival and real-time access.

5

## Visualization: Grafana Dashboards

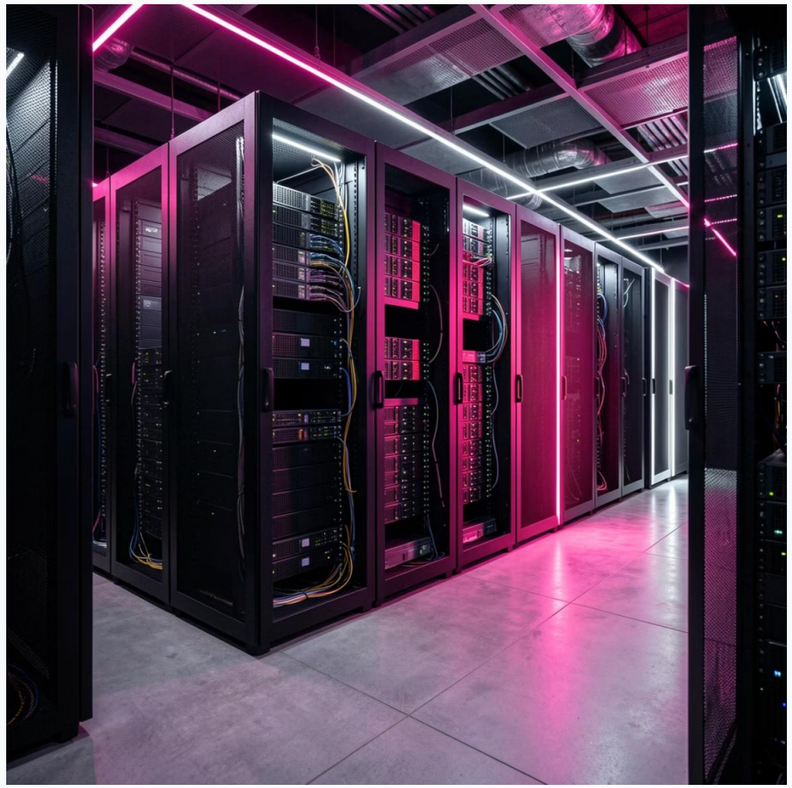
Interactive and insightful data presentation.



# Technology Stack & Infrastructure: Powering the Pipeline

## Cluster Configuration

- 1 Master Node (also functions as a Worker): NameNode, ResourceManager.
- 5 Worker Nodes: DataNodes, NodeManagers.



## Core Technologies

### Orchestration

Hadoop YARN ensures efficient resource management.

### Streaming

Apache Kafka & Spark (PySpark) for real-time data flow.

### Database

PostgreSQL for structured data storage.

### Visualization

Grafana for dynamic and interactive dashboards.

# Data Ingestion Layer: Capturing the Pulse of the Air

## Kafka Producer Implementation

- Polls the OpenAQ API every 60 seconds to capture fresh data.
- Targets key global cities to provide comprehensive coverage.
- Serializes JSON data into a digestible format.
- Pushes processed data to the Kafka topic 'openaq_raw_readings'.
- Decouples data collection from subsequent processing stages, enhancing system resilience.

### Key Monitored Cities

Lahore, Kasur, Faisalabad, Karachi, Peshawar (Pakistan)

Kolkata, New Delhi, Surat, Ajmer (India)

Dhaka (Bangladesh)

Seoul (South Korea)

Hangzhou, Chongqing (China)

Tashkent (Uzbekistan)

Paris (France), London (UK), Zurich (Switzerland), Dublin (Ireland)

Kuala Lumpur (Malaysia)

# Stream Processing Layer: Sparking Insights from Data

## Apache Spark Structured Streaming

Our processing layer leverages Apache Spark Structured Streaming for efficient and reliable real-time data handling.

- **Kafka Topic Consumption:** Continuously reads raw air quality data from the designated Kafka topic.
- **Schema Enforcement:** Skillfully handles complex nested JSON structures, ensuring data integrity and consistency.
- **Data Transformations:** Flattens and cleanses data to prepare it for advanced analysis and storage.
- **Micro-batch Processing:** Guarantees "exactly-once" semantics, preventing data loss or duplication even in distributed environments.

# Storage: The 'Dual Sink' Strategy for Data Resilience

To cater to both immediate and long-term data needs, we implement a robust dual-storage strategy.

## HDFS (Cold Path)

- **Format:** Parquet, a highly efficient columnar storage format.
- **Use Case:** Ideal for historical analysis, big data analytics, and training machine learning models due to its optimized read performance for analytical queries.

## PostgreSQL (Hot Path)

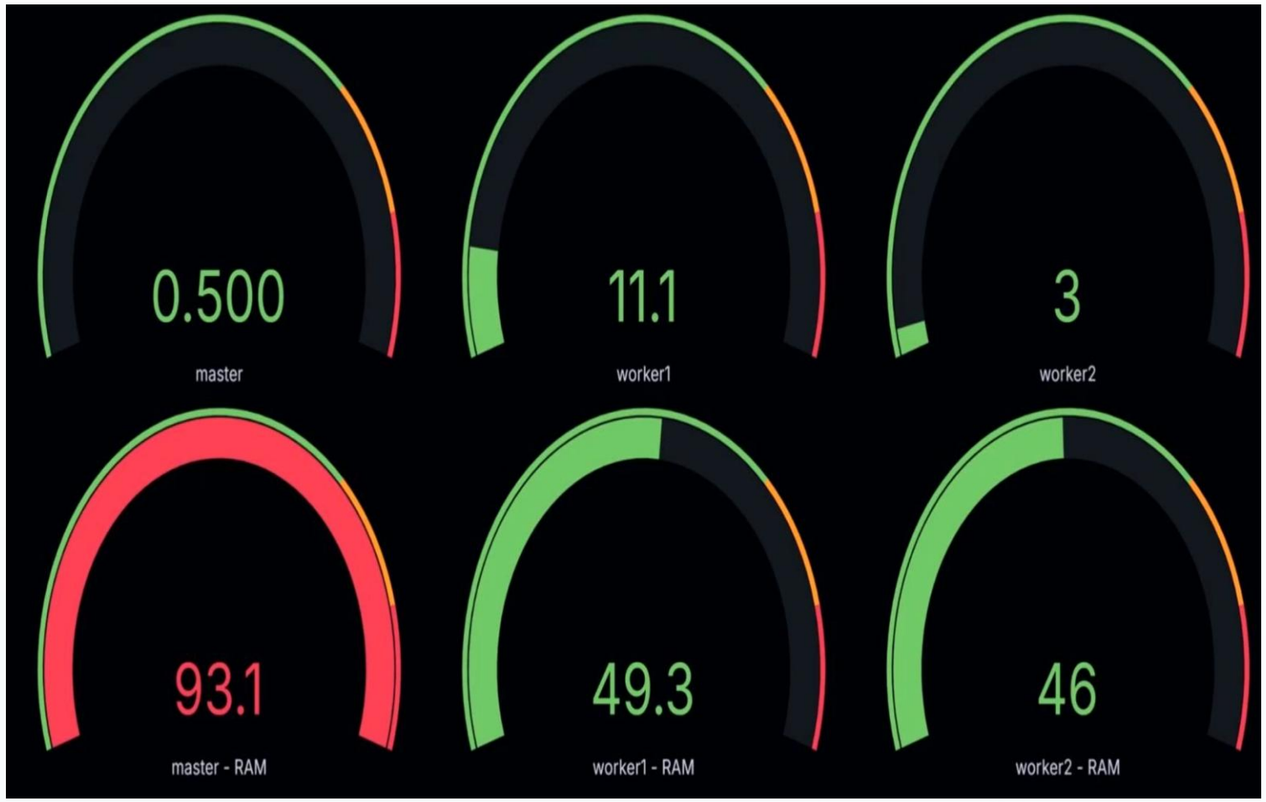
- **Format:** Relational tables, optimized for structured queries.
- **Use Case:** Powers low-latency queries essential for real-time dashboards and immediate data access in Grafana.

# Visualization & Monitoring: Insight at a Glance

## Grafana Integration

Our custom Grafana dashboards provide intuitive and interactive visual representations of air quality data.

- **Geomap:** Offers a spatial overview of pollution levels across monitored cities, identifying hotspots instantly.
- **Time Series:** Enables in-depth trend analysis, revealing patterns such as pollution spikes during rush hours or specific events.




## Cluster Monitoring

Beyond air quality, we continuously track the performance of our distributed cluster, including CPU and RAM usage for both Master and Worker nodes, ensuring optimal system health and resource allocation.

# Live Demonstration: Witnessing Real-Time in Action

We're excited to present a full end-to-end walkthrough of our Real-Time Distributed Air Quality Monitoring System.

## Demo Highlights:

1

**Cluster Startup & Health Check:** Observing the seamless initiation and verification of all nodes.

2

**Spark Job Submission:** Demonstrating the deployment and execution of our Spark streaming jobs.

3

**Real-time Data Flow Verification:** Tracking data as it moves through Kafka, Spark, and into storage.

4

**Dashboard Visualization:** Interacting with live Grafana dashboards to explore dynamic air quality insights.

# Challenges & Future Scope: Evolving the System

## Challenges Encountered

### YARN Memory Allocation

Balancing resource allocation between Spark applications and Hadoop services proved to be a critical optimization task.

### Network Latency

Minimizing delays and ensuring efficient data transfer between geographically distributed nodes required careful fine-tuning.

## Future Scope & Enhancements

### Containerization

Implementing Docker and Kubernetes for enhanced portability, scalability, and easier deployment of the entire system.

### Machine Learning

Developing and integrating ML models to forecast air pollution levels using the rich historical data stored in HDFS.